Lecture Notes in Computer Science 4739

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
University of California, Los Angeles, CA, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Roberto Moreno Díaz Franz Pichler
Alexis Quesada Arencibia (Eds.)

Computer Aided Systems Theory – EUROCAST 2007

11th International Conference
on Computer Aided Systems Theory
Las Palmas de Gran Canaria, Spain, February 12-16, 2007
Revised Selected Papers

Springer

Volume Editors

Roberto Moreno Díaz
Universidad de Las Palmas de Gran Canaria
Instituto Universitario de Ciencias y Tecnologías Cibernéticas
Campus de Tafira, 35017, Las Palmas de Gran Canaria, Spain,
E-mail: rmoreno@ciber.ulpgc.es

Franz Pichler
Johannes Kepler University
Institute of Systems Science
Altenberger Str. 69, 4040 Linz, Austria
E-mail: pichler@cast.uni-linz.ac.at

Alexis Quesada Arencibi
Universidad de Las Palmas de Gran Canaria
Instituto Universitario de Ciencias y Tecnologías Cibernéticas
Campus de Tafira, 35017, Las Palmas de Gran Canaria, Spain
E-mail: aquesada@dis.ulpgc.es

Library of Congress Control Number: 2007939453

CR Subject Classification (1998): J.6, I.6, I.2, J.7, J.3

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-540-75866-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-75866-2 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2007
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12177400 06/3180 5 4 3 2 1 0

Preface

The concept of CAST as Computer Aided Systems Theory was introduced by F. Pichler in the late 1980s to encompass computer-theoretical and practical developments as tools for problem-solving in system science. It was thought of as the third of three components (the other two being CAD and CAM) that together provide a complete picture of the path from computer and systems sciences to practical developments in science and engineering.

Franz Pichler, of the University of Linz, organized the first CAST workshop in April 1988, which demonstrated the acceptance of the concepts by the scientific and technical community. Next, the University of Las Palmas de Gran Canaria joined the University of Linz to organize the first international meeting on CAST (Las Palmas, February 1989), under the name EUROCAST 1989. This proved to be a very successful gathering of systems theorists, computer scientists and engineers from Europe, North America and Japan.

It was agreed that EUROCAST international conferences would be organized every two years. Thus, successive EUROCAST meetings have taken place in Krems (1991), Las Palmas (1993), Innsbruck (1995), Las Palmas (1997), Vienna (1999), Las Palmas (2001), Las Palmas (2003) and Las Palmas (2005), in addition to an extra-European CAST Conference in Ottawa in 1994. Selected papers from those meetings were published in springer's Lecture Notes in Computer Science series, as numbers 410, 585, 763, 1030, 1333, 1798, 2178, 2809 and 3643, and in several special issues of the international journal *Cybernetics and Systems*. EUROCAST and CAST have become well-established meetings, as has been shown by the number and quality of the contributions over the years.

EUROCAST 2007, held in the Elder Museum of Science and Technology of Las Palmas, February 12-16, continued in the same style as previous conferences as an international computer-related conference with a true interdisciplinary character. EUROCAST 2007 included a number of specialized workshops, which were devoted to the following subjects:

- Systems Theory and Simulation, chaired by Pichler (Linz) and Moreno-Díaz (Las Palmas);
- Computation and Simulation in Modelling Biological Systems, chaired by Ricciardi (Napoli);
- Intelligent Information Processing, chaired by Freire (A Coruña);
- Computers in Education, chaired by Martín-Rubio (Murcia);
- Grid Computing, chaired by Volkert (Linz);
- Applied Formal Verification, chaired by Biere (Linz);
- Cellular Automata, chaired by Vollmart (Karlsruhe);
- Computer Vision, chaired by Álvarez (Las Palmas);
- Heuristic Problem Solving, chaired by Affenzeller (Hagenberg);
- Signal Processing Architectures, chaired by Huemer (Erlangen) and Müller-Wipperfürth (Hagenberg);
- Robotics and Robotic Soccer, chaired by Kopacek (Vienna);

- Cybercars and Intelligent Vehicles, chaired by Parent (Paris) and García-Rosa (Madrid);
- Artificial Intelligence Components, chaired by Chaczko (Sidney).

Three invited lectures by Peter Weibel from Karlsruhe, Thomas Arts from Göteborg, and Ricardo García-Rosa from Madrid were complemented with demonstrations on Cybercars by members of the European Cybercar Group (Michel Parent and Ricardo García-Rosa), a presentation on the opportunities for research and experiment in FP7 by Fabrizio Minarini, a lecture on the history of communications, by James Buffalo Kreutzer, and demonstrations of robotic soccer by members of the IHRT Institute of Vienna Technical University, directed by Peter Kopacek.

The papers included in this volume are the result of two successive selection processes: the first for presentation at the conference, and the second a selection of full papers among those presented there. The selection process was the responsibility of the chairpersons, with the counselling of the International Advisory Committee. The present volume has been divided into 13 sections according to the 13 workshops of EUROCAST 2007.

The editors would like to thank all contributors for providing their material in hard and electronic forms on time. Special thanks are due to the Staff of Springer Heidelberg for their valuable support.

July 2007

Roberto Moreno-Díaz
Franz Pichler
Alexis Quesada-Arencibia

Table of Contents

Systems Theory and Simulation: Formal Approaches

On Evolutionary Systems 1

A Systems Theoretic Approach to the Design of Scalable Cryptographic Hash Functions 10

Modelling an Activity in Wireless Sensors Network 18

Explanatory Model for the Break of Logic Equivalence by Irrational Agents in Elkan's Paradox 26

Entropy-Based Modeling and Simulation of Evolution in Biological Systems 34

Logistic Regression as a Computational Tool for Dealing with Intransitivity 42

Synoptic Maps Forecast Using Spatio-temporal Models 50

Ontology-Based Modelling of Session Management Policies for Groupware Applications 57

Computer Aided Development and Simulation of a Remote Control VTOL Tiltwing Model Aircraft 65

A Simulation Study of New Security Schemes in Mobile Ad-Hoc NETworks 73

An Open Modular System for Monitoring Remote Sensors 82

A Structure for Generalized Space-Time Processing: The Feedforward Visual Case 90

Arrays and Continuous Attributes 97

Neuro-Immune-Endocrine (NIE) Models for Emergency Services Interoperatibility 105

Automatic Change Detection in Dynamical System with Chaos Based on Model, Fractal Dimension and Recurrence Plot 113

Picard Discretization of Nonlinear Systems: Symbolic or Numeric Implementation? 121

An Object-Oriented and Generic Compiler Generator 130

A k-NN Based Perception Scheme for Reinforcement Learning 138

Computation and Simulation in Modelling Biological Systems

On the Estimation of First-Passage Time Densities for a Class of Gauss-Markov Processes 146

Simulation of Myosin II Dynamics Modeled by a Pulsating Ratchet with Double-Well Potentials 154

Random Motion with Gamma-Distributed Alternating Velocities in Biological Modeling 163

A Prey-Predator Model for Immune Response and Drug Resistance in Tumor Growth 171

On First-Passage Problems for Asymmetric One-Dimensional Diffusions 179

Multivariate Imputation of Genotype Data Using Short and Long Range Disequilibrium 187

Neonatal EEG Sleep Stages Modelling by Temporal Profiles 195

Intelligent Information Processing

On the Representation of Imperative Programs in a Logical Framework 202

Using Coq to Understand Nested Datatypes 210

Towards a Decentralized and Structured Network of P2P Public Information Screens 217

NowOnWeb: News Search and Summarization 225

Verification of Program Properties Using Different Theorem Provers: A Case Study 233

Temporal Equilibrium Logic: A First Approach 241

A Distributed Filesystem for Spare Storage 249

Generation of Indexes for Compiling Efficient Parsers from Formal Specifications 257

From Text to Knowledge 265

XML Rules for Enclitic Segmentation 273

Certified Genetic Algorithms: Crossover Operators for Permutations 282

Contextual Spelling Correction 290

Multiple Label Text Categorization on a Hierarchical Thesaurus 297

A Formal Foundation for Knowledge Integration of Defficent Information in the Semantic Web 305

Using Temporal Logic for Spatial Reasoning: Temporalized Propositional Neighborhood Logic 313

An Integrated Approach to Filtering Phishing E-mails 321

Formal Reasoning on a Web Coordination System 329

A Multi-agent System for Information Retrieval 337

Knowledge Integration from Multidimensional Data Sources 345

Computers in Education

Evolution of the Design and Implementation of Tutor: A Web-Based Educational System for University Courses 352

Using Videogames in Special Education 360

Facing the Challenges of the European Higher Education Area: The dMath Project 368

A Study of Tool Support for the Evaluation of Programming Exercises 376

Aplying a Semantic Hypermedia Model to Adaptive Concept Maps in Education 384

Towards Virtual Course Evaluation Using Web Intelligence 392

KRRT: Knowledge Representation and Reasoning Tutor System 400

Fuzzy Adaptive Objects (Logic of Monitors as Agents) 408

E-Learning Platform as a Teaching Support in Psychology 415

Wiki Use in Learning for Topography Spanish Students 423

Grid Computing

An Interactive Job Manager for Globus 431

Interactive Molecular Dynamics Simulations on the Grid 443

Extending the Globus Architecture with Role-Based Trust Management 448

GRID Software Solution for the Segmentation of the Coronary Artery Tree in Biplane Angiograms 457

An Intelligent Scatter with Estimation of Distribution for Tabu Search 465

A Framework of GRID Problem-Solving Environment Employing Robust Evolutionary Search 473

A Resources Virtualization Approach Supporting Uniform Access to Heterogeneous Grid Resources 481

Applied Formal Verification

Pattern-Based Verification for Trees 488

Using Verification Technology to Specify and Detect Malware 497

A Compositional Approach for Equivalence Checking of Sequential Circuits with Unknown Reset State and Overlapping Partitions 505

Efficient Model Checking of Applications with Input/Output 515

Ant Colony Optimization for Model Checking 523

On Combining 01X-Logic and QBF 531

Model Checking a Video–on–Demand Server Using McErlang 539

Compressing Propositional Proofs by Common Subproof Extraction 547

Verification of ACTL Properties by Bounded Model Checking 556

Cellular Automata

Application of Linear Hybrid Cellular Automata to Stream Ciphers 564

A Highly Nonlinear Cellular FSM-Combiner for Stream Ciphers 572

Variations on Neighborhoods in CA 581

Hardware Modelling of Cellular Automata: The Game of Life Case 589

Solving the Exploration's Problem with Several Creatures More Efficiently 596

A New Time-Optimum Synchronization Algorithm for Two-Dimensional Cellular Arrays 604

Computer Vision

3D Motion Estimation Using a Combination of Correlation and Variational Methods for PIV 612

Helicopter Flight Dynamics Using Soft Computing Models 621

Movement Identification Analysis Based on 629

Algorithmic Lateral Inhibition Formal Model for Real-Time Motion Detection 638

Second Order Variational Optic Flow Estimation 646

An Application of Optical Flow: Slow Motion Effect on Streaming Image Sequences 654

Comparing Self-calibration Methods for Static Cameras 660

Automation of Snakes in Medical Images 668

Symmetric Optical Flow 676

Real-Time Stereo Visual SLAM in Large-Scale Environments Based on SIFT Fingerprints 684

Tool for Creation Realistic Animation of Human-Like Figures: TRAF 692

An Annotation Tool for Video Understanding 701

Temporal Constraints in Large Optical Flow Estimation 709

Comparing Improved Versions of K-Means and Subtractive Clustering in a Tracking Application 717

FPGA Real Time Lane Departure Warning Hardware System 725

Efficient Combination of the Fuzzy Hough Transform and the Burns Segment Detector 733

Using Fisher Kernel on 2D-Shape Identification 740

Heuristic Problem Solving

Benefits of Plugin-Based Heuristic Optimization Software Systems 747

Metaheuristic Approaches for Optimal Broadcasting Design in Metropolitan MANETs 755

The Influence of Data Implementation in the Performance of Evolutionary Algorithms 764

Heuristic Approach to Conflict Problem Solving in an Intelligent Multiagent System 772

Optimal Placement of Sensors for Trilateration: Regular Lattices vs Meta-heuristic Solutions 780

Selection Pressure Driven Sliding Window Behavior in Genetic Programming Based Structure Identification 788

Multi-Objective Evolutionary Algorithms Using the Working Point and the TOPSIS Method 796

Self-organizing Feature Maps to Solve the Undirected Rural Postman Problem 804

Optimization Methods for Large-Scale Production Scheduling Problems 812

Self-adaptive Population Size Adjustment for Genetic Algorithms 820

Parallel Tabu Search and the Multiobjective Capacitated Vehicle Routing Problem with Soft Time Windows 829

Bandit-Based Monte-Carlo Planning for the Single-Machine Total Weighted Tardiness Scheduling Problem 837

Using GAs to Obtain an Optimal Set of Codes for an Ultrasonic Local Positioning System 845

Using Omnidirectional BTS and Different Evolutionary Approaches to Solve the RND Problem 853

Fast Generation of Production Schedules on a Single Machine 861

Determining Orbital Elements of Extrasolar Planets by Evolution Strategies 870

Training Multiclass Classifiers by Maximizing the Volume Under the ROC Surface 878

Selective Association in Associative Memories – An Extension of the Model NEUNET 886

Towards Constraint-Based Preservation in Systems Specification 894

Automated Re-establishment of the Synchronization of Real and Simulated Project Workflow 903

Automated Modelling in Empirical Social Sciences Using a Genetic Algorithm 912

Extending the Growing Neural Gas Classifier for Context Recognition 920

Signal Processing Architectures

Neural Network Based Path Detection for an FMCW Positioning System 928

Object Oriented Signal Data Structures in VLSI Implementations of Wireless Modems 936

A Bandwidth Efficiency Optimized Frequency Domain Equalization Concept for Single Carrier Transmission 944

Optimized Mapping Schemes for LDPC Coded Higher Order Modulated QAM Transmission 952

Texture-Based Filtering and Front-Propagation Techniques for the Segmentation of Ultrasound Images 960

"Chameleon" Software Defined Control Platform 968

Sampling Rate Conversion for Timing Adjustment in 3.5G Multimode Mobile Terminals 976

Accelerating Space Variant Gaussian Filtering on Graphics Processing Unit 984

Ant-Based Topology Convergence Algorithms for Resource Management in VANETs 992

Simulation of a Signal Arbitration Algorithm for a Sensor Array 1001

Robotics and Robotic Soccer

Mini Robots for Soccer 1009

Embedded Robotic Solution: Integrating Robotics Interfaces with a High-Level CPU in a System-on-a-Chip 1017

An Embedded Vision Sensor for Robot Soccer 1025

MTVS: A Multi-task Active-Vision System 1033

DES-Based Coordination of Space-Sharing Mobile Robots 1041

CASIMIRO, The Sociable Robot 1049

Processing of Myoelectric Signals by Feature Selection and Dimensionality Reduction for the Control of Powered Upper-Limb Prostheses 1057

Cybercars and Intelligent Vehicles

The Anglet Experiment: A Cybercar on the Beach 1066

Crossroad Cooperative Driving Based on GPS and Wireless Communications 1073

Free Space in Front of an Autonomous Guided Vehicle in Inner-City Conditions 1081

Towards a Robust Vision-Based Obstacle Perception with Classifier Fusion in Cybercars 1089

Using Self-adapting Navigation Data for Intelligent, Personalized Vehicle Guidance 1097

Road Approximation in Euclidean and v-Disparity Space: A Comparative Study 1105

Vision-Based Blind Spot Detection Using Optical Flow 1113

Ego-Motion Computing for Vehicle Velocity Estimation 1119

PSPRT: A Case of Pervasive System for Public Road Transport 1126

Study of Correlation Among Several Traffic Parameters Using Evolutionary Algorithms: Traffic Flow, Greenhouse Emissions and Network Occupancy 1134

Robust Obstacle Detection Based on Dense Disparity Maps 1142

Cooperative Maneuver Study Between Autonomous Cars: Overtaking 1151

Trajectory Planning in a Crossroads for a Fleet of Driverless Vehicles ... 1159

Secure and Scalable Communication in Vehicle Ad Hoc Networks 1167

Speed Supervisor for Intelligent Vehicles 1175

Efficient On-Board Stereo Vision Pose Estimation 1183

Artificial Intelligence Components

Complex Software Problem Solving by Means of Abstractive Techniques 1191

Active MIB: Addressing Challenges of Wireless Mesh Networks 1199

Fuzzy Integration of Web Data Sources for Data Warehousing 1208

Comparison of WiFi Map Construction Methods for WiFi POMDP Navigation Systems 1216

A Practical Agent-Based Approach for Pattern Layout Design 1223

Author Index 1229

On Evolutionary Systems

Rudolf F. Albrecht

Faculty of Natural Science, University of Innsbruck, Technikerstr. 25, 6020 Innsbruck, Austria
Rudolf.Albrecht@uibk.ac.at

Abstract. Considered are controllable systems of (structured) objects $s \in S$, S a non-empty set, to which time instants (time "points") $t \in T$ of a partial ordered time $(T, <)$ are assigned. Treated are topological concepts, a theory of controllable variables, ordering relations on $pow\,(T,<)$ induced by $<$, discrete and continuous processes $(s_t)_{t\in U\subset T}$, relations of processes, a general theory of algorithms, neighborhoods of processes, process approximations, and controllable evolutionary processes. Care has been taken of causality and time dependencies of physical processes.

1 Mathematical Prerequisites

1.1 Notations

Notations used are: $pow\ S$ means power set of S; $\leq$ means $<$ or $=$; $\wedge$, $\vee$ means "and", "or"; $\bigwedge$, $\bigvee$ means the universal-, existential quantifier; $\neg$ means "not". $\Rightarrow$, $\Leftarrow$, $\Leftrightarrow$ denote implications; $\cup$, $\cap$ denote set union, set intersection; $\prod$ denotes the (unordered) set product, $\times$ denotes the (finite, ordered) Cartesian set product. If I and M are sets and ι is a function $\iota: I \to M$, we define $m_{[i]} =_{\text{def}} \iota(i)$, $m_i =_{\text{def}} (i, m_{[i]})$, $(m_i)_{i\in I} =_{\text{def}} \{m_i \mid i \in I\}$ is a "family". In this context, I is an "index set", M is an "object set", ι is an "indexing". For convenience sake we use m_I for $(m_i)_{i\in I}$ and (m_i) for $(m_i)_{i\in\{i\}}$ if the meaning is clear from context. For $M = \varnothing$ or $I = \varnothing$ the family is empty. Given a family $M_I = (M_i)_{i\in I}$ of non-empty sets $M_{[i]}$, $R \subseteq \bigcup_{J\subseteq I} \prod_{i\in J} M_i$ defines a relation on M_I, which may be implicitly defined by equivalent "rules" on M_I. If all $M_{[i]} = M$, $\prod_{i\in J} M_i$ is denoted M^J. M_I with relations on it is named "structure" or "system". Independent of an indexing, x_i is a notation.

1.2 Operations on Families

Some important operations on families are:

(1) All set operations like $\cap$, $\cup$, $\subseteq$, considering families as sets. For example: If in a family $(s_i)_{i\in U}$ all $s_{[i]} \in pow\ S$, $V \subseteq U$ and for $i \in V$ $r_{[i]} \subseteq s_{[i]}$, then $(r_i)_{i\in V}$ is a part of $(s_i)_{i\in U}$.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1 – 9, 2007.
© Springer-Verlag Berlin Heidelberg 2007

(2) Given $s_I = (s_i)_{i\in I}$, $U\subseteq I$. Selection of objects by indices, $U \rightarrow \{s_{[i]} \mid s_i\in s_I \wedge i\in U\}$ ("call by name").

(3) Given $s_I = (s_i)_{i\in I}$, $R\subseteq S$. Selection of indices by objects, $R \rightarrow \{i \mid s_i\in s_I \wedge s_{[i]}\in R\}$ ("call by value").
Given $s_U = (s'_i)_{i\in U}$, $s_V = (s''_i)_{i\in V}$, $U,V \subseteq I$.

(4) The projection of s_U on $W\subseteq I$, $pr(W)\, s_U =_{\text{def}} (s_i)_{i\in U\cap W}$.

(5) The concatenation of s_U and s_V, $s_U \kappa s_V =_{\text{def}} (s_i)_{i\in U\cup V}$ with $s_i=s'_i$ for $i\in U\backslash V$, $s_i=s''_i$ for $i\in V\backslash U$. In case $U\cap V \neq \varnothing$, $i\in U\cap V$ and $s'_{[i]}=s''_{[i]}$, then $s_i=s'_i=s''_i$. If $s'_{[i]}\neq s''_{[i]}$, a function $\sigma_i(s'_{[i]}, s''_{[i]})$ is assumed to define $s_i = \sigma_i(s'_{[i]}, s''_{[i]})$. A particular case is $\sigma_i = \mu_i$ with $\mu_i(s'_{[i]}, s''_{[i]}) = s'_i$ if $s'_i = s''_i$. If $U\cap V \neq \varnothing$ and for all $i\in U\cap V$ holds $s'_i = s''_i$, then $s_U \kappa s_V$ is also named composition $s_U \bullet s_V$.

(6) The index differences of s_U and s_V, $s_U\backslash s_V =_{\text{def}} pr(U\backslash V)\, s_U$, $s_V\backslash s_U =_{\text{def}} pr(V\backslash U)s_V$, and the symmetric index difference $\Delta(s_U, s_V) =_{\text{def}} pr(U\backslash V)s_U \cup pr(V\backslash U)s_V$.

These operations can be extended to sets and families of families. For example, the union of $((s_{ij})_{i\in I(j)})_{j\in J}$ yields $(s_{ij})_{(i,j)\in U}$, another decomposition of this is $((s_{ij})_{j\in J(i)})_{i\in I}$, $U = \bigcup_{j\in J} I(j) = \bigcup_{i\in I} J(i)$. Examples for μ are: For sets: union and intersection; for lattices: join and meet; for real numbers: the arithmetic mean $(1/n)(s(1) + s(2) + \ldots + s(n))$, the geometric mean $(s(1) \cdot s(2) \cdot \ldots \cdot s(n))^{(1/n)}$, the harmonic mean $1 / ((1/n)(s(1)^{-1} + s(2)^{-1} + \ldots + s(n)^{-1}))$. If s_U, s_V are elements of a relation R then above operation results need not be elements of R.

1.3 Variables and Their Control

We consider a non-empty family $(r_p)_{p\in P}$ of non-empty structures (i.e. general relations) $r_{[p]}$. For all p let be $r_p = \kappa_p(c, v_{[p]})$, $\kappa_{[p]}$ being a concatenation, which can depend on $(c, v_{[p]})$. Let c be a structure which is independent of all $v_{[p]}$. c can be empty. To facilitate the representation of $R = \{r_{[p]} \mid p\in P\}$ we define as new object the variable var v on variability domain $V = \{(\kappa_{[p]}, v_{[p]}) \mid p\in P\}$ with respect to R, written var v : dom (var v) = (V; R), and variable var r = (var κ) (c, var v) on R. We make the variables "controllable" by associating to var x, $x \in \{r,\kappa,v\}$, a function val with val: (p, var x) $\mapsto x_{[p]}$. The val-functions are named "control-" or "assignment" functions. P, or indirectly any set Q with a given function α: $Q \rightarrow P = \alpha(Q)$, is a set of control- / assignment parameters. For assignments according parameter p we write var x :=(p) $x_{[p]}$. The domain of var x is a set of "states" or "instances" of var x. p itself can be the result of an assignment to a variable var p : P. Thus var v :=(p) $v_{[p]}$, var r :=(p) $\kappa_{[p]}$ (c, $v_{[p]}$).

The variability domain of a variable var v can be structured by relations and can contain variables of lower hierarchical order, to be assigned before an assignment to var v. Variables can be composite and partially assigned, e.g. var v = (var v_1, v_2, v_3). In general, assignments to a part of the variables of a composite variable restrict the assignments to the remaining variable components.

Example 1: Let be defined: var $f : F = \{f_{[p]} \mid p \in P\}$, F being a set of functions $f_{[p]}: X_{[p]} \rightarrow Y_{[p]} = f_{[p]}(X_{[p]})$. We write *var y = var f*(*var x*). Assignment *var f* :=(p) $f_{[p]}$ determines $X_{[p]}$ and $Y_{[p]}$. Next we consider *var* $x_{[p]} : X_{[p]} = \{x_{[pq]} \mid q \in Q(p)\}$ and assign *var* $x_{[p]}$:=(q) $x_{[pq]}$. Then *var y* :=($f_{[p]}$) $f_{[p]}(x_{[pq]})$. According our sequential assignments, *var y* depends on *var x*, *var x* depends on *var f*. If var $x : X \subset X_1 \times X_2$ and if partially assigned var x :=(x_1) var $x' = (x_1$, var $x_2)$, then var x_2 varies on $\{x_2 \mid x_2 \in X_2 \wedge (x_1, x_2) \in X\}$.

1.4 Topological Concepts

We consider a set S and the power set *pow S* of S. (*pow S*, $\cap$,$\cup$) is a complete, complementary, distributive lattice (i.e. Boolean), partial ordered by $\subseteq$ and atomic, the atoms are $\{s\}$, $s \in S$. Let there be given a set $B = \{b_i \mid b_i \in pow\ S \wedge i \in I\}$ with the property: $\bigwedge(b_i, b_j)\bigvee b_k$ $(b_k \subseteq b_i \wedge b_k \subseteq b_j)$, then B defines a "filter base". Dual, if $\bigwedge(b_i, b_j)\bigvee b_k$ $(b_k \supseteq b_i \wedge b_k \supseteq b_j)$, then B defines an "ideal base". Even for a "proper" filter base (i.e. no $b_i = \emptyset$), $\bigcap B = \emptyset$ is possible. If no other neighborhoods are defined and $b^* = \bigcap B \neq \emptyset$, then the filter base is a system of neighborhoods (approximations) to b^*. Dual, for B being an ideal base, if $b^* = \bigcup B \neq \emptyset$, then B is a system of approximations to b^*. For two bases $X = \{x_i \mid x_i \in pow\ S \wedge i \in I\}$, $Y = \{y_j \mid y_j \in pow\ S \wedge j \in J\}$, we define $D = \{x_i \cap y_j \mid (i, j) \in I \times J\}$, $V = \{x_i \cup y_j \mid (i, j) \in I \times J\}$. If X and Y are filter bases, then D and V are filter bases. Dual, if X and Y are ideal bases, then D and V are ideal bases. Important laws are associativity and distributivity.

For filter bases (AssFi): $\bigcap X \cap \bigcap Y = \bigcap D$, (DistrFi): $\bigcap X \cup \bigcap Y = \bigcap V$.

In case $\bigcap D = \emptyset$, but $D \neq \{\emptyset\}$, we define $D^* = \{x_i \cap y_j \mid (i,j) \in I \times J \wedge x_i \cap y_j \neq \emptyset\}$ and assume, $d =_{\text{def}} \bigcap D^* \neq \emptyset$. For $X^* =_{\text{def}} \{x_i \mid x_i \in X \wedge (x_i \mathrm{O} d \neq \emptyset)\}$ and analogue $Y^* =_{\text{def}} \{y_j \mid y_j \in Y \wedge (y_j \mathrm{O} d \neq \emptyset)\}$ we find for the reductions X^*, Y^*, D^* (AssFired): $\bigcap X^* \cap \bigcap Y^* = \bigcap D^* \neq \emptyset$. If $\bigcap X = \emptyset$ in pow S, a completion $\hat{X} \neq \emptyset$ of S may exist such that $\bigcap X = \hat{X}$ in pow $(S \cup \hat{X})$.

For ideal bases (AssId): $\bigcup X \cup \bigcup Y = \bigcup V$, (DistrId): $\bigcup X \cap \bigcup Y = \bigcup D$, which in case $D \neq \{\emptyset\}$ can be similarly reduced to (DistrIdred): $\bigcup X^* \cap \bigcup Y^* = \bigcup D^*$. We consider $\mathbf{C}\bigcup X = \bigcap \mathbf{C}X$. If $\bigcap \mathbf{C}X = \emptyset$ in pow S, a completion $\tilde{X} \neq \emptyset$ of S may exist such that $\bigcap \mathbf{C}X = \tilde{X}$ in pow $(S \cup \tilde{X})$, then $\bigcup X = \mathbf{C}\,\tilde{X}$.

Example 2: Let be: [a, b] $\subset \mathbf{R}$, a<b, f: [a, b] $\xrightarrow{\text{onto}}$ [0, 1] $\subset \mathbf{R}$, $i \in$ [0, 1]. Then $F =_{\text{def}} (\overset{-1}{f}([i, 1])_{i \in [0,\ 1]}$ is a monotone filter base ($\overset{-1}{f}$ means the reciprocal of f), $I =_{\text{def}} (\overset{-1}{f}([0, i])_{i \in [0,\ 1]}$ is a monotone ideal base. Regardless of years earlier existing literature on general topology, by L. Zadeh [10] and many others F was named

"fuzzy set" in contrast to "crisp sets", and a theory of fuzzy sets was developed. Non monotonic filter bases can not be obtained this way, the ideal I was not taken into consideration. By means of two such fuzzy sets on [a, b] and [a´, b´] with "membership functions" f, $f´$ and using for both the same image $([i, 1])_{i\in[0,\ 1]}$, the reduced laws (AssFired), and dual, (DistrIdred) can be illustrated.

If $B = pow\ U$, $C = pow\ V$, $\subseteq$-$\subseteq$- and $\subseteq$-$\supseteq$-homomorphisms $\varphi: B \to C$ can be considered. A particular case of a $\subseteq$-$\subseteq$-endomorphism is a ("rounding", "coarsening") topology $\rho: B \to B$, such that for each $b\in B$ holds $b \subseteq \rho b$, $\rho\rho b = \rho b$. An important application of roundings are "grids": Given a set $I^{(0)} \neq \emptyset$, we consider $A^{(0)} =_{def} \{\{i\} | i\in I^{(0)}\}$ as set of atoms of the lattice $\mathfrak{I}^{(0)} = pow \bigcup A^{(0)}$. A (non identity) rounding $\rho^{(0)} : \mathfrak{I}^{(0)} \to \mathfrak{I}^{(0)}$ is defined by rounding $A^{(0)} : \rho^{(0)}A^{(0)} =_{def} A^{(1)} = \{a^{(1)}{}_i \mid i\in I^{(1)}\}$. From $u \subseteq a^{(1)}{}_i$ follows $\rho^{(0)}\, u = a^{(1)}{}_i$. Let $A^{(1)}$ be the set of atoms of the lattice $\mathfrak{I}^{(1)} = pow \bigcup A^{(1)}$. We have $\mathfrak{I}^{(1)} \subset \mathfrak{I}^{(0)}$. The reciprocals $A^{(0)}(a^{(1)}) =_{def} (\overset{-1}{\rho})^{(0)}(a^{(1)})$ of $a^{(1)} \in A^{(1)}$ form a partition of $A^{(0)}$. We say, $A^{(1)}$ is a "grid" on $A^{(0)}$, the transition from $A^{(0)}$ to $A^{(1)}$ is a coarsening step. Continuing this way, a "multi grid" is generated and the procedure terminates when reaching a singleton set of atoms. The number of steps may be transfinite.

Let $A^{(0)}$ be valuated by elements $s^{(0)}$ of a set $S^{(0)}$, i.e. $(s^{(0)}_a)_{a\in A^{(0)}}$ is given. Let $S^{(1)}$ be a set for valuating $A^{(1)}$. We assume, $\sigma^{(0)}$: $s^{(0)}_{A^{(0)}} \to s^{(1)}_{A^{(1)}}$ is a κ-homomorphism such that $\sigma^{(0)}(s^{(0)}_{A^{(0)}(a^{(1)})}) = (s^{(1)}{}_{a^{(1)}})$. Then for $A\subseteq A^{(1)}$ holds

$$\kappa((\sigma^{(0)}(s^{(0)}_{A^{(0)}(a)}))_{a\in A}) = \sigma^{(0)}(\kappa((s^{(0)}_{A^{(0)}(a)})_{a\in A})).$$

Continuation of this construction results in the generation of a valuated ideal base. In reverse, refining partitions by partitioning and applying correspondences $\overset{-1}{\rho}$, $\overset{-1}{\sigma}$ results in the generation of a valuated filter base.

Example 3: A multi grid to generate an ideal base. $I^{(0)} = [0, 4) \subset \mathbf{R}$, all $S^{(n)} = S = \mathbf{R}$, $A^{(0)} = \{\{i\} | i\in I^{(0)}\}$, $s_{\{i\}} = (e^{-i})_{\{i\}}$, $[i, i+\xi)$ means $\{x \mid i \le x < i+\xi\}$,

$\rho^{(0)}A^{(0)} = A^{(1)} = \{[i, i+1) \mid i\in I^{(1)} = \{0,1,2,3\}\}$, for $\emptyset \neq u \subseteq [i, i+1)$, $\rho^{(0)}u = [i, i+1)$,

$\rho^{(1)}A^{(1)} = A^{(2)} = \{[i´, i´+2) \mid i´\in I^{(2)} = \{0,2\}\}$, for $[i, i+1) \subseteq [i´, i´+2)$, $\rho^{(1)}\,[i, i+1) = [i´, i´+2)$, $\rho^{(2)}A^{(2)} = A^{(3)} = \{[i´´, i´´+4) \mid i´´\in I^{(3)} = \{0\}\}$, for $[i´, i´+2) \subseteq [i´´, i´´+4)$, $\rho^{(2)}\,[i´, i´+2) = [i´´, i´´+4)$. For example, let be

$$\sigma^{(0)}((s^{(0)}{}_{\{i´\}})_{i´\in[i, i+1)}) = (\int_i^{i+1} e^{-i}\,di)_{[i, i+1)} = ((1-e^{-1})e^{-i})_{[i, i+1)} = s^{(1)}{}_{[i, i+1)},$$

$$\sigma^{(1)}((s^{(1)}{}_{[i, i+1)}), (s^{(1)}{}_{[i+1, i+2)})) = (+ ((s^{(1)}{}_{[[i, i+1)]}), (s^{(1)}{}_{[[i+1, i+2)]})))_{[i, i+2)} = s^{(2)}{}_{[i, i+2)},$$

$$\sigma^{(2)}((s^{(2)}{}_{[i, i+2)}), (s^{(2)}{}_{[i+2, i+4)})) = (\max ((s^{(2)}{}_{[[i, i+2)]}), (s^{(2)}{}_{[[i+2, i+4)]})))_{[i, i+4)} = s^{(3)}{}_{[i, i+4)}.$$

Representatives of the structure can be for example the sets $I^{(0)} < I^{(1)} < I^{(2)} < I^{(3)}$ (a "pyramid"). This example illustrates the use of "image pyramids" in Computer Vision (see e.g. [5], [7]).

Example 4: A multi grid to generate a filter base. $\sqrt{2}$ is to approximate in the domain of rational numbers in decimals by nested intervals. The set of valuations is {"0", "1"}, "1" means: by continuity assumption the set $\{\sqrt{2}\}$ is part of the interval. Starting with $\{[i, i+1) \mid i \in \{0,1,2,\ldots 9\}\}$, we find $("1"_{[1, 2)})$. Finer partitioning only this interval, $\{[1.i, 1.i+1) \mid i \in \{0,1,2,\ldots 9\}\}$, we find $("1"_{[1.4,\ 1.5)})$. The grid is coded by the decimal representation 1, 1.4, 1.41, … For the infinite filter base $B = \{[1, 2), [1.4, 1.5), \ldots\}$, $\bigcap B = \varnothing$ in the domain of rational numbers, it is completed by the irrational number $\sqrt{2}$.

2 Times and Processes

2.1 Time

Let T be a finite or infinite set with a non empty irreflexive, asymmetric, transitive order relation $<$. $(T, <)$ represents a "time" with time points, instants $t \in T$. This includes the classical totally ordered time model $(T, <) \subseteq (\boxtimes, \leq)$. Any subset $U \subset T$ is assumed to have the ordering $<$ induced by the ordering $<$ on T. For two subsets $U, V \subset T$, the concatenation $U \kappa V$ in $(T, <)$ is defined by $U \cup V$ with the ordering induced by $<$ on T. For two independent times $(T, <_T)$, $(S, <_S)$, a concatenation is defined by $T \kappa S = (T \cup S, <_{T \cup S})$ with an ordering $<_{T \cup S} = <_T$ on T and $<_{T \cup S} = <_S$ on S, i.e. $<_{T \cup S}$ is compatible with $<_T$ and $<_S$.

$<$ and $\leq$, respectively, on T induce partial directed relations on *pow T*. We consider sub times U, V, W of T. We use $\prec$ in any of the following formulas to represent $<$ or $\leq$, mutually exclusive in this formula. Examples are:

(1) $U \prec_{\wedge(\vee)} V =_{\text{def}} \bigwedge u \in U (\bigvee v \in V (u \prec v))$, from $U \prec_{\wedge(\vee)} V$, $V \prec_{\wedge(\vee)} W$ follows $U \prec_{\wedge(\vee)} W$.

(2) $U \prec_{(\vee)\wedge} V =_{\text{def}} \bigwedge v \in V (\bigvee u \in U (u \prec v))$, from $U \prec_{(\vee)\wedge} V$, $V \prec_{(\vee)\wedge} W$ follows $U \prec_{(\vee)\wedge} W$.

(3) $U \prec_{\wedge(\vee)\wedge} V =_{\text{def}} (U \prec_{\wedge(\vee)} V$ and $U \prec_{(\vee)\wedge} V)$, also denoted by $U \prec V$.
Notice: From $U \leq V$ and $V \leq U$ need not follow $U = V$. Example: U = the interval [0,1] of real numbers, V = the interval [0,1] of rational numbers.

A particular case is: $U <_{\wedge\wedge} V =_{\text{def}} \bigwedge (u,v) \in U \times V\ (u < v)$.

(4) $U \lhd V =_{\text{def}} \neg \bigvee (u,v) \in U \times V\ (v \leq u)$ ("V not before U"). From $U \lhd V$ follows $U \cap V = \varnothing$. In general, $\lhd$ is not transitive. However, from $U \lhd W$, $V \subset W$ and $V \lhd W \backslash V$ follows $U \cup V \lhd W \backslash V$. If $U <_{\wedge\wedge} V$ then $U \lhd V$. If $U \lhd V \wedge V \lhd U$ we say U and V are "parallel", in symbols $U \| V$.

(5) If $U \lhd V$ and $\bigvee (u,v) \in U \times V\ (u < v)$ then we say V depends on, is connected to U with respect to T.

For $U = \{u\}$, $V = \{v\}$, $u \neq v$, these directed relations are equivalent $u < v$. Analogously, directed relations on *pow pow T* etc. can be defined.

A subset $C \subseteq T$ is totally ordered, if for any $t', t'' \in C$ either $t' = t''$, or $t' < t''$, or $t'' < t'$, mutually exclusive. Then C is a chain in T. Any chain C is element of T^C. For $t \in T$, $\{(t)\}$ is a chain. $(T, <)$ is the union of all its chains.

2.2 Processes and Process Relations

Let Z be a set , U be a sub time of $(T, <)$, S be *pow Z*. Any $s \in S$ can be structured, this means relations r can be defined on it. A "process on S over U" is defined by $(s_t)_{t\in U}$, i.e. a time function. $<$ on U induces an ordering of the process elements: $s_t < s'_{t'}$ if and only if $t < t'$. Notice: If $s_t < s'_{t'}$, then $s_{[t]} = s'_{[t']}$ is possible. $s_{[t]}$ is a "state" of s_U. We name a change of a state $s_{[t]} \rightarrow s'_{[t']}$, $t<t'$, an "event" with respect to (t, t'). The above ordering relations (1) to (4) induce ordering relations on processes.

To model physical processes, we take causality and operational delay into account. According our physical experience, if a cause at time instant t_{cause} results in an effect at time instant t_{effect}, we assume $t_{cause} < t_{effect}$. If an operation acts on an object *obj*(1) at $t(1)$ and the operation results in an object *obj*(2), then the physical generation of *obj*(2) generates a time point $t(2)$ with $t(1) < t(2)$. If this $<$ relation is not contained in $(T, <)$, we extend $<$ on T by this.

Considering two processes $s_U = (s'_t)_{t\in U}$, $s_V = (s''_t)_{t\in V}$ with $U \cap V \neq \emptyset$ we say, the processes are "concurrent" on $U \cap V$ with respect to T.

All previous reasoning on operations on families, variables, topology can be transferred to the special case of times as index sets. This includes hierarchies, homomorphisms, roundings and neighborhood systems of processes.

Example 5: Neighborhoods of a process. On real numbers we consider the set of processes $\{(s(\delta,\varepsilon)_t \mid t\in[0,1] \wedge \delta,\varepsilon \in \mathbf{N}/10 \wedge (s(\delta,\varepsilon)_{[t]} = ([t-\delta,\ t+\delta] \times [t^2-\varepsilon,\ t^2+\varepsilon])\}$. For example, the set $F = \{(s(\delta,\varepsilon)_t)_{\ t\in[0,1]} \mid \delta \geq 0.1,\ \varepsilon \geq 0.1\}$ forms a filter base as neighborhood system to $\bigcap F = (s(0.1,0.1)_t)_{\ t\in[0,1]}$, for example the process $(s(0.3,0.1)_0, s(0.2,0.3)_{0.5}, s(0.4,0.1)_1)$ is in neighborhood $(s(0.4,0.3)_t)_{\ t\in[0,1]}$.

Two processes $s_U = (s_t)_{t\in U}$, $r_V = (r_t)_{t\in V}$, $U, V \subseteq T$, can have an interaction: Let there be given functions φ_t: $s_t \rightarrow r'_{t'} \subseteq r_{t'}$, $\psi_{t''}$: $r''_{t''} \rightarrow s'''_{t'''} \subseteq s_{t'''}$, then by our assumptions $t<t'$, $t''<t'''$. $r'_{t'}$ is "input" to r_V from s_U , $r_{t''}$ is "output" to s_U from r_V. φ_t, $\psi_{t''}$ are "transfer" functions ("export"-, "import" functions). Their application may change their argument objects. For time compatibility, if $t'<t''$ and $(t, t''') \in <$, then $t<t'''$. We assume, s_U represents a physical process and r_V represents a physical environment of s_U. Then we may consider an "observation", "recognition", "measurement" of a state of s_U as output of s_U. States of s_U not influenced or created by input or output are "endogenous", otherwise "exogenous".

Let s_U and $\Delta(s_U) =_{\text{def}} (\Delta s)_{\Delta U} = ((\Delta s)_t)_{t\in\Delta U} \neq \emptyset$ be processes on U and ΔU, respectively; $U, \Delta U \subseteq T$; $s_{[t]}, (\Delta s)_{[t]} \in S$. If $s_U \triangleleft \Delta(s_U)$, i.e. $U \triangleleft \Delta U$, then by definition of $\triangleleft$, $U \cap \Delta U = \emptyset$. $s_U\ \kappa\ \Delta(s_U)$ is a "continuation" of s_U . Of special interest are increments $\Delta(s_U)$ "closest" (in a sense defined in the following) to s_U.

The considered objects like $t\in T$, T, $s\in S$, S, φ_t, ψ_t, $\Delta(s_U)$ can be variables and then have to be replaced by admissible values for a particular application.

2.3 Algorithms

Let there be given a time $(T, <)$ and a sub time $(U, <) \subset (T, <)$. We assume

A1: $\overline{U} \subseteq T\backslash U$, $\overline{U} \neq \emptyset$, and $\overline{U}$ is a maximal set in T with $U \lhd \overline{U}$.

A2: $\Delta U \subseteq \overline{U}$, $\Delta U \neq \emptyset$ (for infinite sets a selection applies the "axiom of choice").

A3: $\Delta U \lhd \overline{U} \backslash \Delta U$.

From this follows:

$\neg\bigvee(t\in T\backslash \overline{U})(U <_{(\vee)\wedge} \{t\})$ by A1. $\neg\bigvee(t\in \overline{U} \backslash \Delta U)(\{t\} <_{\wedge(\vee)} \overline{U} \backslash \Delta U)$ by A3. Hence $\neg\bigvee(t\in T)(U <_{(\vee)\wedge} \{t\} <_{\wedge(\vee)} \Delta U))$, $\bigwedge(t',t''\in \Delta U)((t\in T \wedge t'<t<t'') \Rightarrow (t\in \Delta U))$.

ΔU is "neighboring" U in T, in symbols $U << \Delta U$. $\overline{U} \setminus \Delta U$ is a maximal set in T with $(U \kappa \Delta U) \lhd (\overline{U} \setminus \Delta U)$. $U \kappa \Delta U$ is an "evolution" (or closest continuation) of U in T. If two times $(U, <_U)$, $(V, <_V)$ are parallel, then $U \cup V$ is an evolution of U and of V. For two families $U = (U_i)_{i\in I}$, $V = (V_i)_{i\in I}$ with $(U_i \lhd V_i)_{i\in I}$ holds

(P1) $\kappa U \lhd \bigcap V$, (P2) $\bigcap U \lhd \kappa V$, especially for $\lhd$ replaced by $<<$. In this case (and usually for finite I) (P1) expresses the computation of the "Pareto Front" in optimization.

Given a valuation of a time $T^{(o)}$ by elements of S, $(s_t)_{t\in T}{}^{(0)}$. If $T^{(o)}$ is $<<$-"well ordered", we set $T^{(o)} = T$, if not, we apply a rounding ρ: pow $T^{(o)} \xrightarrow{\text{onto}} T \subset$ pow $T^{(o)}$ and apply a homomorphism $\sigma^{(0)} : S \to S$ according the first coarsening step in a grid, such that T is $<<$-well ordered. The image is denoted by $(s_t)_{t\in T}$. This enables the following "algorithm" on $(s_t)_{t\in T}$:

Initial values: $U = \emptyset$, $\overline{U} = T$, $i = 0$, $\Im = \{0\}$, these quantities seen as variables.

(1) Choose $\Delta(s_U) \subseteq (s_t)_{t\in \overline{U}}$ such that $\emptyset \neq \Delta U \subseteq \overline{U}$, $U<<\Delta U<<(\overline{U} \setminus \Delta U)$,

(2) assign $s_U := s_U \kappa \Delta(s_U)$, $\overline{U} := \overline{U} \setminus \Delta U$,

(3) assign $\Im := \Im \cup \{i+1\}$,

(4) if $\overline{U} = \emptyset$ terminate, else go to (1).

$\Im$ or an equivalent set determines the "algorithmic time" of the algorithm, which can be transfinite. $U =_{\text{def}} (U_i)_{i\in\Im}$ forms an ideal base, $\bigcup U = T$, $\overline{U} =_{\text{def}} (\overline{U}_i)_{i\in\Im}$ forms a filter base (Fréchet), $\bigcap \overline{U} = \emptyset$, the function $(s_t)_{t\in T}$ is "filtered" (is a Moore-Smith sequence). If S is a topological space, limits of the sequence may exist. The steps on grids (Section 1.4) are examples of algorithms.

2.4 Controllable Evolutionary Processes

For an algorithm we assume the computational steps of states s were performed up to algorithmic time i, $(s_{T(s)})_i$, and the next incremental step is a variable $(\text{var } \Delta(s_{T(s)}))_i$ (this denotes $((\text{var } \Delta s)_{\text{var } \Delta T(s)})_i$). $(\text{var } \Delta(s_{T(s)}))_i$ is assumed to be the concatenation of an exogenous part $(\text{var } \Delta(s'_{T(s')}))_i$ and an endogenous part $(\text{var } \Delta(s''_{T(s'')}))_i$. The variables vary on a parameterized domain ΔW_i of admissible continuations, parameterized by an incremental control process $(\text{var } \Delta(c_{T(c)}))_i$, continuation of the past control process $(c_{T(c)})_i$. ΔW_i depends on the past state- and control processes and possible constraints. The endogenous part is a function of past processes and the incremental control process variable. Formulated for step i:

$(\text{var } \Delta(s_{T(s)}))_i = ((\text{var } \Delta(s'_{T(s')}))\ \kappa f(s_{T(s)}, c_{T(c)}, \text{var } \Delta(c_{T(c)})))_i$,

assigned by $\Delta(c^*_{T(c^*)}) = (\Delta(s'_{T(s')}), \Delta(c_{T(c)}))$:

(EvStates) $(\Delta(s_{T(s)}))_i = (\Delta(s'_{T(s')})\ \kappa f(s_{T(s)}, c_{T(c)}, \Delta(c_{T(c)})))_i$, in integral form

$(s_{T(s)})_{i+1} = (s_{T(s)})_i\ \kappa(\Delta(s_{T(s)}))_i$ (states depend on control).

Times: $(T(s))_i << (\Delta T(s))_i$ by causality, operation delay and newly *created* $(\Delta T(s))_i$, $(T(c))_i < (\Delta T(c))_i < (\Delta T(s))_i$. Dual to (EvStates) is

(EvControl) $(\Delta(c_{T(c)}))_i = ((\Delta(c'_{T(c')})\ \kappa g(s_{T(s)}, c_{T(c)}, \Delta(s_{T(s)})))_i$, in integral form

$(c_{T(c)})_{i+1} = (c_{T(c)})_i\ \kappa(\Delta(c_{T(c)}))_i$ (control depends on states).

In topological spaces and under further conditions, limits with respect to algorithmic time yield differential and integral forms of the evolution laws.

Example 6: $(s_t = s'_t + \int_1^t (ds''/dt)d\tau)_{t\in[1,2)} = ((s'_{[t]} - s''_{[1]})_t + s''_t)_{t\in[1,2)}$, $(c_t = ((s'_{[t+1]} - s''_{[1]})_t,\ ((ds''/dt)_{[t+1]})_t))_{t\in[0,1)}$. (EvStates): $(s_t = (c'_{[t-1]})_t + \int_1^t c''_{[\tau-1]}d\tau)_{t\in[1,2)}$.

(EvControl): $(c_t = ((s'_{[t-1]} - s''_{[1]}), (ds''/dt)_{[t-1]})_t)_{t\in[2,3)}$.

Example 7: Computer Architectures, Manufacturing Systems. Let $(s_{T(s)})_i$, $(c_{T(c)})_i$ be given (physically stored). All objects are processes, in the following time subscripts are suppressed. Set $f_i = (f_{il})_{l\in L(i)}$, f_{il}: $x_{il} \mapsto \Delta s''_{il}$, $\Delta s''_i = \kappa((\Delta s''_{il})_{l\in L(i)})$ and let $\tilde{x}_{il}$ be a selected part of $(s_{T(s)})_i$ which is time isomorphic x_{il}, and $\tilde{x}_i = \kappa((\tilde{x}_{il})_{l\in L(i)})$. $T(\tilde{x}_{il}) << T(x_{il}) <_{\wedge(\vee)\wedge} T(\Delta s''_{il})$. For given $\Delta s'_i$ let be defined $\Delta s_i = \Delta s'_i\ \kappa \Delta s''_i$. In case of a manufacturing system $s_{i+1} = (s_i \setminus \tilde{x}_i)\ \kappa \Delta s_i$, in case of a computer system $s_{i+1} = s_i\ \kappa \Delta s_i$. Notice: The selection of $\tilde{x}_i$ is subject to domain constraints, $x_i \in \text{dom } f_i$, and capacity constraints, $\tilde{x}_i \subseteq s_i$.

Application examples are:

(1) Classical architecture: $L(i)$ has only one element, $\Delta s'_i$ contains a synchronizing clock process, selection of f_i, $\tilde{x}_i$ by the operating (control) system according an imperative programming language.
(2) Multi-processor architecture: Like (1), but $L(i)$ has more than one element.
(3) Pipelining is included, because we admit overlapping processes.

(4) Data flow architectures (see e.g. [3]): Asynchronous selection of the next computation step is performed according a given program structure and availability of data and processors ready to compute.
(5) Neural network architectures: For layer 1: $L(1)$ processors are provided to classify $L(1)$ pattern classes by "learning" a neighborhood system for each class and afterwards using these to classify a sample and process the result in following layers 2,3,… Classically, a set of linearly bounded neighborhoods is used; later Gaussian type neighborhoods were reinvented and used.
(6) Processors processing neighborhoods as estimates for uncertainty, e.g. interval arithmetic, "fuzzy numbers", minimized rounding arithmetic (see e.g. [4,8]).

3 Conclusions

We tried to present a comprehensive and unifying theoretical framework covering most of the principles used in constructive mathematics, numerical analysis, controlled computational and manufacturing systems. Treated were processes on partial ordered time and emphasized were the topological foundations of successive approximations and uncertainty. The axioms A1, A2, A3 we postulated for partial ordered sets are in the literature on unstructured sets (see e.g. [6]), our concept of well-ordering generalizes the classical one. Our scheme for algorithms on partial ordered sets with finite or infinite many steps reduced to finite graphs is known (see e.g. [9]). Hints at applications of our approach are given in the Examples. Parts of this article were treated in earlier publications of the author, e.g. [1,2].

References

1. Albrecht, R.F.: On mathematical systems theory. In: Albrecht, R. (ed.) Systems: Theory and Practice, pp. 33–86. Springer, Vienna-New York (1998)
2. Albrecht, R.F: Topological Interpretation of Fuzzy Sets and Intervals. Fuzzy Sets and Systems 135, 11–20 (2003)
3. Gaudiot, J.L., Bic, L. (eds.): Advanced Topics in Data-Flow Computing. Prentice Hall, Englewood Cliffs, N.J. (1991)
4. Herzberger, J.: Einführung in das wissenschaftliche Rechnen. Addison-Wesley-Longman, Redwood City, CA, USA (1997)
5. Jolion, J.M., Rosenfeld, A.: A Pyramid Framework for Early Vision. Kluwer Academic Publishers, Dordrecht (1994)
6. Klaua, D.: Mengenlehre, Walter de Gruyter Berlin-New York (1979)
7. Kropatsch, W.G.: Properties of Pyramidal Representations. In: Kropatsch, W., Klette, R., Solina, F. (eds.) Theoretical Foundations of Computer Vision, Computing Suppl., vol. 11, pp. 99–111. Springer, Vienna-New York (1996)
8. Kulisch, U.W., Miranker, W.L.: Computer Arithmetic in Theory and Practice. Academic Press Inc, New York (1981)
9. Roy, B.: Algébre et théorie des graphes. Dunod Paris (1969)
10. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

A Systems Theoretic Approach to the Design of Scalable Cryptographic Hash Functions

Josef Scharinger

Johannes Kepler University, Institute of Computational Perception,
4040 Linz, Austria
Josef.Scharinger@jku.at

Abstract. Cryptographic hash functions are security primitives that compute check sums of messages in a strong manner and this way are of fundamental importance for ensuring integrity and authenticity in secure communications. However, recent developments in cryptanalysis indicate that conventional approaches to the design of cryptographic hash functions may have some shortcomings.

Therefore it is the intention of this contribution to propose a novel way how to design cryptographic hash functions. Our approach is based on the idea that the hash value of a message is computed as a message-dependent permutation generated by very special chaotic permutation systems, so called Kolomogorov systems. Following this systems theoretic approach we obtain arguably strong hash functions with the additional useful property of excellent scalability.

1 Introduction and Motivation

Cryptographic hash functions for producing checksums of messages are a core primitive in secure communication. They are used to ensure communication integrity and are also essential to signature schemes because in practice one does not sign an entire message, but the cryptographic checksum of the message.

All the cryptographic hash functions in practical use today (SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512) are specified in the Secure Hash Standard (SHS, see [8]) and are based on ideas developed by [illegible] for his MD5 message digest algorithm [9]. Unfortunately, recent attacks [14] on SHA-1 show that this design approach may have some shortcomings. This is the reason why the intention of this contribution is to deliver a radically different systems theory based approach to the design of scalable cryptographic hash functions.

The reminder of this contribution is organized as follows. In section 2 we explain the notion of a cryptographic hash function. Section 3 introduces the well-known class of continuous chaotic Kolomogorov systems, present a discrete version of Kolomogorov systems and analyze cryptographically relevant properties of these discrete Kolomogorov systems. Next, section 4 describes our novel approach to the design of cryptographic hash functions which is essentially based on the idea of computing a message check sum as a message dependent permutation generated by iterated applications of the discrete Kolmogorov systems

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 10–17, 2007.
© Springer-Verlag Berlin Heidelberg 2007

described in section 3. Finally, section 5 intends to justify the claim that our design of cryptographic hash functions based on systems theory constitutes a highly scalable approach to the development of cryptographic hash functions.

2 Cryptographic Hash Functions

2.1 The Concept of a Cryptographic Hash Function

Following [11], cryptographic hash functions come under many different names: one-way hash function, message digest function, cryptographic checksum function, message authentication code, and quite some more. Essentially a cryptographic hash function takes an input string and converts it to a fixed-size (usually smaller) output string.

In a more formal way, a cryptographic hash function $H(M)$ operates on an arbitrary-length plaintext (message) M and returns a fixed-length hash value $h = H(M)$, where h is of length N. While one can think of many functions that convert an arbitrary-length input and return an output of fixed length, a cryptographic hash function has to have additional characteristics:

- [illegible]: given M, it is easy to compute h, but given h, it is hard to compute M
- [illegible]: given M, it is hard to find another message M', such that $H(M) = H(M')$ and even more it should be hard to find two arbitrary messages M_1 and M_2 such that $H(M_1) = H(M_2)$

It is perfectly obvious to see that any cryptographic hash function producing length N hash values can only offer order $\mathcal{O}(2^N)$ security with respect to fulfilling the one-way property. Even more, taking into consideration the so-called [illegible] [11], it follows that any cryptographic hash function can only offer order $\mathcal{O}(2^{N/2})$ security with respect to collision resistance. It is therefore essential to note that N defines an upper limit on security that is achievable by any length N cryptographic hash function. Accordingly it would be nice to have [illegible] hash functions where increasing N should be as simple as possible, a point we pay special attention to with our approach presented in this paper.

3 Chaotic Kolmogorov Systems

Among the most remarkable results of recent systems theory are novel findings on chaotic systems. There has been good progress in systems science concerning the analysis of complex dynamical systems and concepts like fractal dimension or strange attractors are now well understood. However, it is worth noting that the overwhelming majority of exiting systems is by definition of continuous type, so system states are in some power set of $\mathbb{R}$.

A fundamental property of chaotic systems is the fact that small deviations in inputs can completely alter the systems behavior. This immediately leads to

the problem that any approximations, as inherently involved by any digitization, may change systems behavior completely. Therefore, for practical digital applications of interesting chaotic systems it is essential to successfully bridge the gap from continuous type systems to discrete version that still preserve the essential properties present in the continuous case.

In our contribution we focus on the class of chaotic Kolmogorov systems [3,6,13]. This class has been of great interest to systems scientists for a long time due to some unique properties amongst which the outstanding degree of instability is particularly remarkable. It has been proven [2] that continuous Kolmogorov systems T_π guarantee ergodicity, exponential divergence and perfect mixing of the underlying state space for almost all valid choices of parameter π. Note that these properties perfectly match the properties of [illegible] (as first defined by [illegible] in [12]) that are so fundamental in cryptography.

3.1 Continuous Kolmogorov Systems

Continuous chaotic Kolmogorov systems act as permutation operators upon the unit square $\mathbb{E}$. Figure 1 is intended to give a notion of the dynamics associated with a specific Kolmogorov system parameterized by the partition $\pi = (\frac{1}{3}, \frac{1}{2}, \frac{1}{6})$. As can be seen, the unit square is first partitioned into three vertical strips according to $\frac{1}{3}, \frac{1}{2}, \frac{1}{6}$. These strips are then stretched to full width in the horizontal and squeezed by the same factor in the vertical direction and finally these transformed strips are stacked atop of each other. After just a few applications (see Fig. 1 from top left to bottom right depicting the initial and the transformed state space after 1, 2, 3, 6 and 9 applications of T_π) this iterated stretching, squeezing and folding achieves excellent mixing of the elements within the state space.



Fig. 1. Illustrating the chaotic and mixing dynamics associated when iterating a Kolmogorov system

Formally this process of stretching, squeezing and folding is specified as follows. Given a partition $\pi = (p_1, p_2, \ldots, p_k)$, $0 < p_i < 1$ and $\sum_{i=1}^{k} p_i = 1$ of the unit interval $\mathbb{U}$ and stretching and squeezing factors defined by $q_i = \frac{1}{p_i}$. Furthermore, let F_i defined by $F_1 = 0$ and $F_i = F_{i-1} + p_{i-1}$ denote the left border of the vertical strip containing the point $(x, y) \in \mathbb{E}$ to transform. Then the continuous Kolmogorov system T_π will move $(x, y) \in [F_i, F_i + p_i) \times [0, 1)$ to the position

$$T_\pi(x, y) = (q_i(x - F_i), \frac{y}{q_i} + F_i). \tag{1}$$

It is well known and proven [2] that [illegible] π the corresponding continuous Kolmogorov system T_π fulfills the following appealing properties:

- [illegible]: guarantees that almost any initial point approaches any point in state space arbitrarily close as the system evolves in time. Speaking in terms of cryptography this property can be considered as equivalent to [illegible] since initial (input) positions does not give any information on final (output) positions.
- [illegible]: neighboring points diverge quickly at exponential rate in horizontal direction. Speaking in terms of cryptography this property can be considered as equivalent to [illegible] since initially similar initial (input) positions rapidly lead to highly different final (output) positions.
- [illegible]: guarantees that all subspaces of the state space dissipate uniformly over the entire state space. Speaking in terms of cryptography this property can be considered as a perfect equivalent to [illegible].

Deducing from this analysis it can be concluded that continuous Kolmogorov systems offer all the properties desired for a perfect permutation operator in the [illegible] domain. Our task now is to develop a [illegible] version of Kolmogorov systems that preserves these outstanding properties. That is precisely what will be done in the next subsection.

3.2 Discrete Kolmogorov Systems

In our notation a specific discrete Kolmogorov system for permuting a data block of dimensions $n \times n$ shall be defined by a list $\delta = (n_1, n_2, \ldots, n_k)$, $0 < n_i < n$ and $\sum_{i=1}^{k} n_i = n$ of positive integers that adhere to the restriction that all $n_i \in \delta$ must partition the side length n.

Furthermore let the quantities q_i be defined by $q_i = \frac{n}{n_i}$ and let N_i specified by $N_1 = 0$ and $N_i = N_{i-1} + n_{i-1}$ denote the left border of the vertical strip that contains the point (x, y) to transform.

Then the discrete Kolmogorov system $T_{n,\delta}$ will move the point $(x, y) \in [N_i, N_i + n_i) \times [0, n)$ to the position

$$T_{n,\delta}(x, y) = (q_i(x - N_i) + (y \bmod q_i), (y \operatorname{div} q_i) + N_i). \tag{2}$$

As detailed in the preceding subsection, continuous Kolmogorov systems T_π are perfect (ergodic and mixing) permutation operators in the continuous domain. Provided that our definition of discrete Kolmogorov systems $T_{n,\delta}$ has the same desirable properties in the discrete domain, that would deliver a strong permutation operator inherently possessing the properties of confusion, diffusion and perfect statistics in the sense that permutations produced are statistically indistinguishable for truly random permutations. The analysis in the next subsection proofs exactly that this is true indeed.

3.3 Analysis of Discrete Kolmogorov Systems

As detailed in [10], the following theorem can be proven for discrete Kolmogorov systems T_{n,δ_r}:

Theorem 1. ... $n = p^m$ ... p. ... T_{n,δ_r} ... $4m$ ... δ_r ... r ... $4\log_2 n$ ...

For any cryptographic system it is always essential to know how many different keys are available to the cryptographic system. In our case of discrete Kolmogorov systems $T_{n,\delta}$ this reduces to the question, how many different lists $\delta = (n_1, n_2, \ldots, n_k)$ of n_i summing up to n do exist when all n_i have to part n?

As detailed in e.g. [1], a computationally feasible answer to this question can be found by a method based on formal power series expansion leading to a simple recursion relation. If $R = \{r_1, r_2, \ldots, r_m\}$ denotes the set of admissible divisors in ascending order, then c_n, the number of all lists δ constituting a valid key for $T_{n,\delta}$, is given by

$$c_n = \begin{cases} 0, & \text{if } n < r_1 \\ c_{n-r_1} + c_{n-r_2} + \ldots + c_{n-r_m} & \text{if } (n \geq r_1) \wedge (n \notin \{r_1, r_2, \ldots, r_m\}) \\ 1 + c_{n-r_1} + c_{n-r_2} + \ldots + c_{n-r_m} & \text{if } n \in \{r_1, r_2, \ldots, r_m\} \end{cases} \tag{3}$$

Some selected results are given in table 1. To fully appreciate these impressive numbers note that values given express the number of permissible keys for just one round and that the total number of particles in the universe is estimated to be in the range of about 2^{265}.

4 Hash Functions from Chaotic Kolmogorov Systems

Deducing from theorem 1, the following holds true:

- if random parameters δ_r are used and at least $4\log_2 n$ rounds are iterated, then any $n \times n$ square will be perfectly permuted by applying a sequence of transformations T_{n,δ_r}
- this permutation is determined by the sequence of parameters δ_r

Table 1. Number of permissible parameters δ for parameterizing the discrete Kolmogorov system $T_{n,\delta}$ for some selected values of n

n	c_n	n	c_n	n	c_n
4	1	8	5	16	55
32	5.271	64	47.350.055	128	$\approx 2^{50}$
256	$\approx 2^{103}$	512	$\approx 2^{209}$	1024	$\approx 2^{418}$

This immediately leads to the following idea how to calculate the hash value for a message M using discrete Kolmogorov systems T_{n,δ_r}:

- the bits of the message M can be interpreted as a sequence of parameters δ_r
- the application of a sequence of transforms T_{n,δ_r} will result in a permutation hash determined by message M

According to this principle, our algorithm for the calculation of a [illegible] of length N for a message M works as described next.

4.1 Initialization

In the initialization phase all that has to be done is fill a square array of side length n (such that $n \times n = N$) with e.g. left half $N/2$ zeros and right half $N/2$ ones.

4.2 Message Schedule

Next we partition message M into blocks M_i (e.g. of size 512 bits). This is useful e.g. because this way the receiver of a message can begin calculation of hash values without having to wait for receipt of the entire message and additionally this keeps our approach in compliance with e.g. the HMAC algorithm [7] which demands an iterated hash function in its very definition.

Then we expand block M_i to get a corresponding expanded pseudo-random message block W_i. This can e.g. be done using linear congruence generators (LCGs, see [5]), linear feedback shift registers (LFSRs, see [4]) or the expansion mechanisms used in the Secure Hash Standard (SHS, see [8]) defined by NIST. All we demand is that this expansion has to deliver $\geq 4 \log_2 n$ M_i-dependent pseudo-random groups $g_{i,r}$ of bits interpretable as parameters $\delta_{i,r}$ (see following two subsections on interpretation and use of bit groups $g_{i,r}$).

4.3 Mapping Bit Groups onto Partitions

When the task is to map bit groups $g_{i,r}$ from M_i and W_i onto valid parameters $\delta_{i,r} = (n_1, n_2, \ldots, n_k)$ this can be accomplished in very simple ways. Just examine the following illustration:

M_i or W_i			
$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	...
0 1 ... 0	1 1 ... 0	0 1 ... 1	.....
$\delta_{i,1}$	$\delta_{i,2}$	$\delta_{i,3}$	...

If one writes down explicitly the different partitions $\delta_{i,r}$ possible for various n (whose total number is given in Tab. 1) one immediately notices that the probability of a value n_i being contained in a partition decays exponentially with the magnitude of n_i. Therefore the following approach is perfectly justified. When iterating over bit groups $g_{i,r}$ and generating factors n_i for $\delta_{i,r}$, we interpret the smallest run of equal bits (length 1) as the smallest factor of n, namely $n_i = 2^1 = 2$, a run of two equal bits as factor $n_i = 2^2 = 4$, and so on.

There are two details to observe in the above outlined procedure of mapping bit groups $g_{i,r}$ onto valid partitions $\delta_{i,r}$:

- One point to observe in this procedure is that the sum of n_i's generated from bit groups $g_{i,r}$ this way has to equal n. Therefore one has to terminate a run as soon as an n_{j+1} would be generated such that $\sum_{i=1}^{j} n_i + n_{j+1} > n$. Then the maximum possible n_{j+1} (as a power of 2) still fulfilling the constraint has to be chosen, and the run length has to be reset to one.
- The other point to observe is that one iteration over $g_{i,r}$ may yield n_i summing up to less than n. In that case $g_{i,r}$ just has to be scanned iteratively until the n_i generated sum up to n indeed.

Observance of these two details will guarantee that valid parameters $\delta_{i,r}$ will always be generated for bit groups $g_{i,r}$.

4.4 Processing a Message Block

Processing message block M_i and corresponding expanded message block W_i is perfectly straightforward. Just iteratively take groups of bits $g_{i,r}$ first from M_i then from W_i, map them onto parameters $\delta_{i,r}$, permute square array according to $T_{n,\delta_{i,r}}$, and finally rotate the array by $g_{i,r}$ mod N to avoid problems with fixed points $(0,0)$ and $(n-1,n-1)$. All one has to care about in this simple scheme is that groups $g_{i,r}$ taken from M_i must have sizes k, such that 2^k is lower or equal to the number of permissible keys (see Tab. 1) for $T_{n,\delta_{i,r}}$ to avoid collisions, and that groups $g_{i,r}$ taken from W_i must have sizes k, such that 2^k is greater or equal to the number of permissible keys for $T_{n,\delta_{i,r}}$ to ensure perfect mixing according to theorem 1.

Applying this procedure for all message blocks M_i of message M will result in excellent chaotic mixing of the square array [illegible] M.

4.5 Reading Out the Message Digest

Finally, reading out the state of array reached after processing all M_i yields a strong checksum of length $N = n \times n$ for message M.

5 Scalability

Some readers might wonder why our description of Kolmogorov permutation hashes as specified in section 4 does not fix a specific value N for the length of hash values produced by our approach. The reason is simple: we want our approach to the design of cryptographic hash functions to be as generic as possible. As already indicated in the title of this contribution, we are aiming at the development of . . . cryptographic hash functions.

To understand why this scalability is so important, recall from section 2 that it is a fact that an N bit hash function can only offer security up to level $\mathcal{O}(2^{N/2})$ [11]. Consequently, as computing power is increasing steadily, it may become desirable to increase the length of hash values produced without having to redesign the hash function.

In our scheme, increasing the length and thus achieving remarkable scalability is straightforward. By just changing the size of the underlying square array from $n \times n$ to $2n \times 2n$, the length of hash values produced is increased by 4. Obviously, this involves minor modifications to block expansion and bit group partitioning as explained and specified in section 4, but besides these small changes, the same algorithm can be used.

References

1. Aigner, M.: Kombinatorik. Springer, Heidelberg (1975)
2. Arnold, V.I., Avez, A.: Ergodic Problems of Classical Mechanics. W.A. Benjamin, New York (1968)
3. Goldstein, S., Misra, B., Courbage, M.: On intrinsic randomness of dynamical systems. Journal of Statistical Physics 25(1), 111–126 (1981)
4. Golomb, S.W.: Shift Register Sequences. Aegan Park Pr., Laguna Hills, CA (1981)
5. Knuth, D.E.: The Art of Computer Programming. Addison-Wesley, London, UK (1998)
6. Moser, J.: Stable and Random Motions in Dynamical Systems. Princeton University Press, Princeton (1973)
7. NIST. Keyed-Hash Message Authentication Code (HMAC). FIPS 198 (March 2002)
8. NIST. Secure hash standard (SHS). FIPS 180-2 (August 2002)
9. Rivest, R.L.: The MD5 message digest function. RFC 1321 (1992)
10. Scharinger, J.: An excellent permutation operator for cryptographic applications. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 317–326. Springer, Heidelberg (2005)
11. Schneier, B.: Applied Cryptography. Addison-Wesley, London, UK (1996)
12. Shannon, C.E.: Communication theory of secure systems. Bell System Technical Journal 28(4), 656–715 (1949)
13. Shields, P.: The Theory of Bernoulli Shifts. The University of Chicago Press, Chicago (1973)
14. Wang, X., Yin, Y.L., Yu, H.: Finding collisions in the full SHA-1. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, Springer, Heidelberg (2005)

Modelling an Activity in Wireless Sensors Network

Jan Nikodem

The Institute of Computer Engineering, Control and Robotics
Wrocław University of Technology
11/17 Janiszewskiego Street, 50-372 Wrocław, Poland
jan.nikodem@pwr.wroc.pl

Abstract. In this paper we describe the results of modelling an activity in WSN; we concentrate on effects of organising groups of sensors, on relationships between sensors and the effects on their autonomy. Consequently our work are concentrated on developing the formal methods and techniques necessary to model and evaluate situations in the network, processes of decisions making and implementing intelligent behavior while following the general outlines of the network activity. These requirements will entail investigating ways in which intelligent systems evaluate, interact and self-organise, both individually and cooperating with other aerial explorers or interacting with environmental stimulus.

Keywords: wireless sensors network, teleogenic systems, holons.

1 Introduction

A wireless sensors network (WSN) is a set of homogeneous sensors which interact with each other. For identification purpose each sensor has a unique ID number. Apart from where it interacts with other sensors, it is autonomous entity and its behavior strongly depends on challenges caused by the environment. The interactions between sensors are realised mainly in information domain, so an essential point is: how to provide appropriate communication capabilities. The huge number of papers has already been carried out on this subject, such as [17,20] which proposed different protocols that manages the communication. It is also worth to mention papers [19,22] devoted to self-organising during the clustering and routing time and its range in wireless sensors network. Broadly speaking, these approaches pointed at clustering and multi-hop routing methods as crucial elements that guarantee decentralised energy efficient realisation of mentioned above aims. These approaches prolong network lifetime by distributing energy.

Proposed in this paper approach is slightly different from mentioned above. A wireless sensors network is much more than simple network of unsophisticated technical elements performing tasks defined by engineers. We are considering wireless sensor network as a set of sensors deployed in some area, which must act autonomously and cooperatively to achieve their particular self-interested and the global goals. Each sensor has restricted energy, hardware and software

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 18–25, 2007.
© Springer-Verlag Berlin Heidelberg 2007

resources or communication range, so it is not capable to realise individually overall system tasks. Because of that, sensors need to have more or less tightly integrated cooperation (based on information exchange) between them. This cooperation results in collective behavior of unsophisticated sensors interacting locally with their environment. Moreover, such cooperation causes coherent functional global reactions which fulfils network performance requirements.

This raises new challenges in establishment of sensors coalitions/clusters and their cooperation - autonomy balance inside obtained clusters. Building sensors coalition results in recursively aggregated cooperation domains which are ordered hierarchically. The components of each level of this hierarchy should fulfil a specific requirements and realise adequate to this level tasks (neighbourhoods - network discovering, cluster - environment monitoring, routing tree - communication channel etc,). Such attempt leads us to hierarchical, adaptive systems described in [2,3,12]

Cooperation concept will not be useful only for describing relationships on some level; it will be very useful for multi-layer hierarchies - situations where delegation and autonomy are necessary because a base station or cluster head or another delegator does not have the local, decentralised and updated data, or the just-in-time reactivity, or some physical skill that requires some local resources. In that sense, cooperation is an activity realised both on each layer as well as across multi-layer structure.

2 Cybernetical Approach for Wireless Sensors Network

There are a vast number of ideas founded at 50s and developed through the sixties and the seventies of twenty century related to Systems Science and second order cybernetics. Some of these concepts have been assimilated by other disciplines. Other ideas seem to have been forgotten, though to be periodically reinvented in different domains. Nowadays, thanks to information and computer technologies development, we may use the power of modern computers to simulate and thus experiment with and develop some of the well defined ideas. Two of them; Koestler's idea of holonic [10] and Jaroń 's approach to teleogenic systems [8] we adopted as a conceptual framework for the purpose of our attempt.

2.1 Holarchical Concept for Wireless Sensor Network

Holonic systems were proposed by Arthur Koestler [10]. The main idea is the fact that, in real environment, an entity should be considered both as a component of a set and as whole made up of other entities. A holon defined as being a part of a whole should fulfilled three conditions [1]:

- *to be stable.* Stability means that a holon is able to cope and react when it is subjected to high demand or to major disturbances,
- *to have a capacity for autonomy.* Autonomy suggests that a holon is capable of managing itself when subjected to demands in order to achieve its own aims,

– *to be capable of cooperating.* The capability to cooperate means that the holons are able to coexist with other holons or other layers of holons, and are capable of working on common aims.

Concerning the first characteristic - stability, we presented our proposal in [9] taking into account such situations as detection of sensors misbehaviour or illegal sensor intrusions. Approach proposed in [9] allows interacting and negotiating between sensors in cluster. Such systems have increased network immunity based on adjustable autonomy [5,6,16], swarm intelligence [21] or an artificial immune system [18]. When sensors in cluster detect changes in their environment or detect deviations from the normal system behaviour they negotiate the common reaction according to the formula: local event, local communication, local decision, local action and thus reduce the impact of a failure in any single component. In this approach the stability of the whole network is more important than the stability of each of its sensors. So, it is sometimes necessary that some sensor be temporarily separated or eliminated from community [4], thus allowing the network to can take more long-term protection strategies. In such situation the stability results in overall network robustness (survivability and dependability).

In this paper we concentrate on remaining two of the holon characteristics. Moreover in our attempt we look for the specification of a holonic wireless sensors network which offers a good compromise between the autonomy and cooperation. Sensors in WSN compose a different type of coalitions like clusters or routing trees. Each cluster head as Koestler's holon has "Janus-face property" [11]. When 'looking down' the aspect of inside cluster communication is very important (autonomy tendency), but 'looking up' cluster head as a member of routing tree integrates its functions according to routing requirements (integration tendency) [14]. Besides, by his nature, a cluster heads like each other regular sensors has particular self-interested goal - minimizing energy consumption [22].

2.2 Teleogenic Systems and Goal-Space Relations in WSN

In this chapter we focus on formal model of teleogenic complex cybernetical system described by Jaroń [8]. He augments the complex of traditional attempts and notions up to axiomatically defined cybernetical systems, their special mappings: functionisms and morphisms as well as their architecture and behaviors. Jaron's idea of "teleologics" based on the best background of mathematical logic, topology and set theory is extremely useful in our attempt, especially in aspects of such fundamental concept like goals their space and relation determined over that space. Starting from this point a WSN will be treated as a specific class of complex cybernetical system. Thus, according to Jaron's concept, three types of relation are necessary in WSN:

– *tolerance relation.* Described relationships between some sensors which can build a coalition sharing a common goal, but concurrently they realise their own objectives and bilaterally tolerate each other activity. The tolerance relation is a pure mathematical concept, so this relation is reflexive and symmetrical,

– *subordination relation.* Subordination is binary relation between some sensors which can partially resign from their autonomy and merge into new holon. In this relation one sensor dominates another. Subordination by its nature is reflexive and transitive, consequently, it is an ordering relation defined on the WSN goal-space,
– *collision relation.* It express that merging into new holon, self-interested sensors retaining its autonomy on certain aspects and prefer what is beneficial to them according to anticipated profits. Collision relation by its nature is symmetrical and irreflexive.

There are some additional dependencies between these three mentioned above relations which will be very useful in our considerations. The collision and tolerance relations as well as collision and subordination are disjoint. Finally some very useful property should be mentioned, that subordinated activity must tolerate all activities tolerated by dominants.

Two of mentioned above relation (tolerance and collision) describe activity realised both on each layer and across multi-layer structure when the last of them - subordination works only between two or more layers. Subordination relation reflects hierarchical structure of dependency in WSN but in slight different meaning than Koestler's looking-up integration or Ferber's cooperation. We situated our approach between these two concepts. Instead of multi-strata hierarchy [13] or six functions associated to local components [7] we propose three relations well defined over the WSN goal space. In such approach integration requires to minimise the collisions. Cooperation results from balance between growing up subordination and tolerance when autonomy results from balance between decreasing subordination and growing up collision.

3 Modelling an Activity in Wireless Sensors Network

3.1 Basic Notions

Describing WSN activity we will be talking about actions, trajectories and behaviour. *Action* will be considered as a property of any network elements like sensor, cluster head or node. On the contrary, the *behaviour* is an external attribute which should be considered either as an attribute of all WSN or its subset (cluster, tree, vicinity). **Action:** is a ternary relation defined as follows:

$$Action : Sensors \times State \rightarrow Sensors. \tag{1}$$

Terms: describe necessary and sufficient conditions for doing something. The terms are suitable for describing ($\Rightarrow$) the interaction between environment and WSN elements. Terms usually reflected environmental boundaries of WSN elementary actions.

$$(\forall Act \in Actions)\,(\exists term \subseteq Terms)\,(term \Rightarrow Act)\,. \tag{2}$$

Rules: they are action descriptors. Rules carry information how to realise ($\succ$) action associated with them, therefore:

$$(\forall Act \in Actions)\,(\exists rule_{ORD} \subseteq Rules)\,(rule_{ORD} \succ Act) \tag{3}$$

where $rule_{ORD}$ is a subset of rules which is partially or totally ordered. Thus for any two rules from $rule_{ORD}$ either one precedes another, or is its successor, otherwise they are equivalent to each other.

Notice, that any action should have corresponding, ordered subset of rules determining the realisation of this action.

Trajectories: it is evident that there are many subsets of *Actions*, but from our point of view, some are more interested than another. We are looking for a specific subsets which elements form time dependent chain, as follows:

$$Action_t\,(Sens, State_t) = Action_t\left(Sens, Action_{(t-1)}\left(Sens, State_{(t-1)}\right)\right) \tag{4}$$

where $Action_t$ is successor of $Action_{(t-1)}$.

The above consideration leads us to define a formal notion of chains of a length m on a set *Actions*; their set will be denoted by $Actions^{(m)}$. In order to construct the sets $Actions^{(m)}$ for any length m, we will use the well-known operation of concatenation. The definition is formulated inductively [8]:

1. $Actions^{(0)} = \{empty\ word\}$
2. $Actions^{(1)} = Action$
3. If the elements of $Actions^{(m)}$ are defined, then
   $Actions^{(m+1)} = \{x, y \mid < x, y > \in Actions^{(m)} \times Action\}$.

The set of chains on the set *Actions* will be denoted by $Actions^{(*)}$. Defined below chains of the ordered actions, varying at time are described as relations included in a time space, we will be called ***Trajectories***

$$Actions^{(*)} = \bigcup \{Actions^{(m)} \mid m \in \mathbb{N}\}. \tag{5}$$

Behaviour: is an another notion related to *Actions* we use in further consideration. In order to define behaviour in the WSN lets consider a partition of the set *Actions* with respect to relation $\mathcal{R}$, where the relation is an equivalence relation on *Actions*. Based on this we can construct the quotient set, whose elements are called equivalence classes with regard to the relation $\mathcal{R}$, denoted as:

$$Behaviour = Actions/\mathcal{R} = \{Act \in Actions \mid \mathcal{R}\,(Act)\}. \tag{6}$$

Such a definition of WNS behaviour can not raise any doubts. Now it is obvious that WNS behaviour includes nothing more than its actions: real and possible.

3.2 Actions and Behaviour in WSN

Concerning the WSN in a time domain we can describe networks activity based on notions presented in previous chapter. All network lifetime proceeding from

preprocessing to bootup, maturity, a degraded phase, and finally to network failure and death, will be considered based on topological, time-space related objects, as sets, classes and relations.

In this chapter we tend to portray WSN lifetime activity by the implementation of actions and behaviour concepts. Taking into account actions and behaviour in a time domain, the first should be considered as time stamped but behaviour is always related to time period. An actions belonging to each quotient set (behaviour) are arranged into time dependent chains but on the other side, actions are always associated with some sensor. During the lifetime of a network we may distinguished such phases as:

- ***preprocessing***. The efficiency of the WSN activity heavily depends on accuracy of information which is programmed into a sensors during the factory cycle. This is a time when engineer's objective is to design a set of actions and describe the rules related to them. The terms and its permissible ranges are defined too. Programmers investigate and design the appropriate computational form for these various issues, and determine how these rules and terms should mutually relate to realise a consistent whole. Preprocessing creates the ability for any sensor to realise elementary actions as an component of the network as a whole, but in this stage the network exists only virtually in engineers and programmers visions,
- ***power-up initialization***. Upon power-up, each sensor executes some initialization routines, such as internal self-test and built-in calibration or status determination. It also launches some procedures that have been preprogrammed to reflect specific mission requirements and expectations,
- ***self-organization*** is the first true WSN behaviour. The ability of each sensor to self-organise is essential to the success of the sensors network. Sensors which are deployed on some area started to cooperate with each other creating communication and functional network structure. Each sensor discovers its vicinity searching other sensors as well as the environment characteristic. As a result of this activity each sensor has its own view of the world around it, and maintains its own database embedded in this node. This behaviour could be repeated many times relating to maximise the sensor system performance and lifetime,
- *network discovery*. When the sensor becomes a member of network it should next determine its topological relationship within the vicinity,
- *neighbours localization*. Once the topology surrounding the sensor is known, it should learn the neighbourhood structure and neighbour's position localization due to the known range limitations,
- *networking* is a sub-behaviour. The sensor determines whether it can hear an already-operational sensor network, by listening for invitations to join and possibly "overhearing" other ongoing communications,
- *clustering*. Due to the fact that sensors are left unattended clustering is distributed and this decisions based process uses local information only. Cluster heads are responsible for inter-cluster and inside-cluster communication so their selection has to be based mainly on an energy level of each sensor. We need to be sure that a strongest nodes will be selected,

- *routing*. The communication is the crucial aspect of the WSN activity. Thus there is a critical need for the routing to balance traffic particularly near the sink nodes, using energy as the primary performance metric. The routing algorithms should select the appropriate paths through the network to extend the network lifetime and provide its connectivity as long as possible.
- ***working*** is the leading behaviour of WSN.
- ***dusking*** is the last behaviour in network life. Traditionally it has to have three phases: challenging, exhausting and surviving.

4 Conclusions and Perspectives

In order to formally specify an activity in Wireless Sensors Network, we turned towards conceptual framework coined by Koestler [10] and Jaroń [8]. Based on these fundamental notions we can describe, in formal manner, the notions of actions and behaviour. If the sets *Actions* are not clearly determined, then in order to describe behaviour as the choice from among them, we should be able to evaluate each *Action* and know the interrelation among them.

These notions will be very useful in further work. It seems to be a good starting point to determine cooperation and autonomy abstractions and consider it in the context of three basic relations (subordination, tolerance, collision). As a term *Action* is strictly connected with network elements (e.g. sensors) an autonomy and cooperation terms are related to some community in the network (e.g. clusters, communication trees). Based on relations described above we can define autonomy as property of some community. The scope of autonomy is determined by decreasing subordination and growing up collision relations when scope of cooperation is determined by growing up subordination and tolerance.

We belief, that using this concept based on rigorous mathematical approach may leads us to very fruitful results. Proposed, formal mathematical approach allows us to simulate and thus experiment with and develop such exciting ideas like swarm or distributed intelligence, adjustable autonomy or autonomy-cooperation dichotomy.

References

1. Adam, E., Mandiau, R., Kolski, C.: HOMASCOW: A Holonic Multi-Agent System for Cooperative Work, DEXA Workshop, pp. 247–253 (2000)
2. Castelfranchi, C.: Founding Agent's Autonomy on Dependence Theory. In: proceedings of ECAI 2000, Berlin, pp. 353–357 (2000)
3. Cerpa, A., Estrin, D.: ASCENT: Adaptive Self-Configuring Sensor Networks Topologies. IEEE Transactions On Mobile Computing 3(3) (2004)
4. Chaczko, Z., Ahmad, F.: Wireless Sensor Network Based System for Fire Endangered Areas, ICITA 2005, Sydney (2005)
5. Crandall, J.W., Goodrich, M.A.: Experiments in adjustable autonomy. In: IEEE International Conference on Systems, Man, and Cybernetics, Tucson, USA, vol. 3, pp. 1624–1629 (2001)

6. Falcone, R., Castelfranchi, C.: The Human in the Loop of a Delegated Agent: the Theory of Adjustable Social Autonomy. IEEE Trans. on Systems, Man and Cybernetics, 406–418 (2001)
7. Ferber, J.: Multi-Agent Systems. An Introduction to Distributed Artificial Intelligence. Addison Wesley, London (1999)
8. Jaroń, J.: Systemic Prolegomena to Theoretical Cybernetics, Scient. Papers of Inst. of Techn. Cybernetics, Wrocław Techn. Univ., no. 45, Wrocław (1978)
9. Klempous, R., Nikodem, J., Radosz, Ł., Raus, N.: Adaptive Misbehavior Detection in Wireless Sensors Network Based on Local Community Agreement. In: 14th Annual IEEE Int. Conf. and Workshops on the Engineering of Computer-Based Systems, ECBS 2007, Tucson, USA, pp. 153–160 (March 2007)
10. Koestler, A.: The Ghost in the Machine. Hutchinson & Co. (1967)
11. Koestler, A.: Janus. A Summing Up. Hutchinson & Co Ltd, London (1978)
12. Lin Ch, R., Gerla, M.: Adaptive Clustering for Mobile Wireless Networks. IEEE Journal On Selected Areas In Communications 15(7) (1997)
13. Pichler, F.: Modeling Complex Systems by Multi-Agent Holarchies. In: Kopacek, P., Moreno-Diaz, R., Pichler, F. (eds.) EUROCAST 1999. LNCS, vol. 1798, pp. 154–168. Springer, Heidelberg (2000)
14. Pichler, F.: Holonic Peripheral Systems Components: A General Systems Approach. In: Shafazand, H., Tjoa, A.M. (eds.) EurAsia-ICT 2002. LNCS, vol. 2510, pp. 43–49. Springer, Heidelberg (2002)
15. Scerri, P., Pynadath, D., Tambe, M.: Towards Adjustable Autonomy for the Real World. Journal of Artificial Intelligence Research 17 (2003)
16. Schillo, M.: Self-organization and adjustable autonomy: Two sides of the same medal? Connection Science 14(4), 345–359 (2003)
17. Sohrabi, K., Gao, J., Ailawadhi, V., Pottie, G.J.: Protocols for Self-Organization of a Wireless Sensor Network. IEEE Personal Communications (2000)
18. Su P., Feng D.:The Design of an Artificial Immune System, Int. Conf. on Networking. In: Systems and Mobile Communications and Learning Technologies (2006)
19. Vaidya, D., Peng, J., Yang, L., Rozenblit, J.W: A Framework for Sensor Management in Wireless and Heterogeneous Sensor Network. In: ECBS'2005, 12th IEEE International Conference on the Engineering of Computer-Based Systems, Greenbelt, USA, 4-7 April, pp. 155–162 (2005)
20. Veyseh, M., Wei, B., Mir, N.F.: An Information Management Protocol to Control Routing and Clustering in Sensor Networks. Journal of Computing and Information Technology - CIT 13(1), 53–68 (2005)
21. Veeramachaneni, K., Osadciw, L.: Dynamic Particle Swarm Optimizer for Information Fusion in Non Stationary Sensor Networks, IEEE Swarm Intelligence Symposium, Indianapolis, USA (2006)
22. Younis, O., Fahmy, S.: HEED: A Hybrid, Energy-Efficient,Distributed Clustering Approach for Ad Hoc Sensor Networks. IEEE Transactions On Mobile Computing, 3(4) (2004)

Explanatory Model for the Break of Logic Equivalence by Irrational Agents in Elkan's Paradox

Germano Resconi[1] and Boris Kovalerchuk[2]

[1] Dept. of Mathematics and Physics, Catholic University, via Trieste 17, Brescia, Italy
resconi@numerica.it
[2] Dept. of Computer Science, Central Washington University, Ellensburg, WA 98922, USA,
borisk@cwu.edu

Abstract. Fuzzy logic breaks logic equivalence of statements such as $(A\wedge B)\vee(\neg A\wedge B)\vee(A\wedge\neg B)$ and $A\vee B$. It breaks the symmetry of use of such logically equivalent statements. There is a controversy about this property. It is called a paradox (Elkan's paradox) and interpreted as a logical weakness of fuzzy logic. In the opposite view, it is not a paradox but a fundamental postulate of fuzzy logic and one of the sources of its success in applications. There is no explanatory model to resolve this controversy. This paper provides such a model using a vector/matrix logic of rational and irrational agents that covers scalar classical and fuzzy logics. It is shown that the classical logic models rational agents, while fuzzy logic can model irrational agents. Rational agents do not break logic equivalence in contrast with irrational agents. We resolve the paradox by showing that the classical and fuzzy logics have different domains of rational and irrational agents.

Keywords: Fuzzy logic, classical logic, explanatory model, logic equivalence, irrational agent, inconsistent agent, rational agent, paradox.

1 Introduction

In classical set theory the union of two sets **A** and **B** is equal to the union of their intersection $\mathbf{A}\cap\mathbf{B}$ with their proper parts, $\mathbf{A}\backslash\mathbf{B}$ and $\mathbf{B}\backslash\mathbf{A}$, i.e., $\mathbf{A}\cup\mathbf{B} = (\mathbf{A}\cap\mathbf{B}) \cup (\mathbf{A}\backslash\mathbf{B}) \cup (\mathbf{B}\backslash\mathbf{A})$. In classical logic, this property means logic equivalence of truth-values of appropriate statements, $A \vee B = (A\wedge B) \vee (\neg A\wedge B) \vee (A\wedge\neg B)$, but in fuzzy logic this is not true in general,

$$A \vee B \neq (A\wedge B) \vee (\neg A\wedge B) \vee (A \neg B).$$

Fuzzy logic *breaks logic equivalence*. We cannot use $(A\wedge B) \vee (\neg A\wedge B) \vee (A\wedge\neg B)$ instead of $A \vee B$. In other words, fuzzy logic *breaks logic symmetry* of use of these statements. For the first time, the problem of interrelations between equivalence in classic and in fuzzy logic was, discussed in [2] for classical equivalence between disjunction and conjunction normal forms, DNF and CNF.

It is shown in [2] and [5] that not only the DNF and CNF are different in fuzzy logic, but also any classic equivalence is substituted with a class of alternative relations that are not equivalences.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 26 – 33, 2007.
© Springer-Verlag Berlin Heidelberg 2007

There is a controversy about this break of symmetry property in the literature [6], [7], [13],[14]. It is even called a paradox (Elkan's paradox) and interpreted as a logical weakness of fuzzy logic because it does not follow the classical logic equivalence and related properties. The alternative view is that it is not a paradox, but a fundamental postulate of fuzzy logic and one of the sources of its success in applications. None of the sides provided an *explanatory model* to justify these contradictory views beyond the arguments presented above.

Elkan's theorem says that if we assume $\neg(A \wedge \neg B) = B \vee (\neg A \wedge \neg B)$ for all sentences A and B in standard min/max fuzzy logic, then for any propositions P and Q it follows that the truth value of P is either the same as the truth value of Q or its negation. This means that if fuzzy logic would follow this classical equivalence it will be useless at best.

Our goal is to give an explanatory model to shed light on this issue. The model uses a concept of a *logic of uncertainty* of rational and irrational agents that includes both classical logic, fuzzy logic and other logics. We state that classical logic can model only rational agents, but fuzzy logic can model irrational agents too. Rational agents do not break logic equivalence, but irrational agents do this.

We resolve the controversy about equivalence by showing that classical logic and fuzzy logic have different domains of rational and irrational agents, respectively. It does not mean that there is no overlap, where both theories model rational agents, but this issue is beyond the scope of this paper.

2 Explanatory Model

2.1 The Scope of Logic

To analyze the controversy we start from the basic question: "What is logic?' Obviously, there are many possible answers, but it is important for our analysis to stress that logic is a reasoning mechanism on *abstracted sentences*. Abstraction means that out of the infinite number of properties {p} of sentence S in a natural language only few properties $\{p_{abs}\}$ are associated with an abstracted sentence S_A. Now we need to distinguish between *logic* and *model* of the real world situation. Logic can be viewed as a model of a real word expressed in abstracted sentences.

Models that use less expressive languages may produce less realistic model of the world. For instance, the propositional logic is much more limited than the first order logic (FOL). However, the classic logic has no goal to build a comprehensive model of a real world situation. By definition, the classic propositional logic tries to construct truth-values of some sentence from *only truth-values* of other sentences, not from semantic *contents* of sentences themselves. It seems not accidental that the probability theory is not called a logic. The probability theory is not limited by this assumption.

The classical logic uses a very limited semantics of sentences. Logics of uncertainty should use much more semantics to be able realistically model complex uncertain situations [12]. Involving much more semantic information implies that the situation and a logic that models the situation become interdependent. They have a

high overlap in the set parameters. The question is “How to incorporate semantics to the logic of uncertainty?”

In this paper, we show that the use of a mixture of rational and irrational agents is a way to incorporate important semantic information.

2.2 Elkan’s Paradox and Its Meaning as a Break of Symmetry in Logic

The proof of the Elkan’s theorem [6] is based on the *assumption of classical equivalence applied to the fuzzy logic*. Specifically, it uses classic equivalence $\neg(A\wedge\neg B)= B\vee(\neg A\wedge\neg B)$, which does not hold in fuzzy logic with A and B from [0,1] and operations, $A \wedge B = \min(A,B)$, $B \vee A = \max(A, B)$, $\neg A =1– A$.

It raises the questions: “Should we require the classical logic equivalence of sentences for every logic of uncertainty? “What should logic of uncertainty be?” If the classical equivalence is not required and justified in fuzzy logic then Elkan’s paradox is not a paradox. It is proposed in [2] that another relation, which is a kind of similarity relation, should substitute the classical equivalence in fuzzy logic. Given membership function μ with only two values $\{0,1\}$ the classic logic holds properties:

(1) $\mu (X \wedge Y) = \mu (X) \cdot \mu (Y)$
(2) $\mu (X \vee Y) = \mu (X) + \mu (Y) - \mu (X) \mu (Y)$
(3) $\mu (\neg X) = 1 - \mu (X)$
(4) $\mu (X \wedge Y) = \mu (X) - \mu (\neg Y \wedge X) = \mu (Y) - \mu (\neg X \wedge Y)$
(5) $\mu (X \vee Y) = \mu (X) + \mu (Y \wedge \neg X) = \mu (Y) + \mu (X \wedge \neg Y)$
(6) $\mu (\neg X) = \mu (\neg X \oplus IR)$,

where operation “$\oplus$ “ is XOR and $IR= \neg X \wedge X$. In classical logic, IR is always false, IR=0. Therefore, $\mu(\neg X)=1 \Rightarrow \mu(\neg X \oplus IR)=1$, and $\mu(\neg X)=0 \Rightarrow \mu(\neg X \oplus IR)=0$. Thus, in classical logic adding IR using XOR to any statement X keeps logic equivalence. Below we show that this is not case for the irrational agents that allow $\neg X \oplus IR$ to be true.

2.3 Reasoning with Conflicting and Irrational Evaluations

As a departure from the classical logic, the proposed logic of uncertainty allows the proposition $\neg X\wedge X$ to be not false, $\mu(\neg X\wedge X) > 0$, in accordance with the evaluation provided by an irrational agent. If this happened, we call $\neg X\wedge X$ an irrational sentence. We will also call any other sentence as an *irrational sentence/statement* and denote it as IR if it takes a value that is impossible in the classical logic. For instance, we have $\mu (X\vee\neg X)=1$, that is a classical logic tautology, but an irrational agent may claim that $\mu (X\vee\neg X)<1$. In this case, $X\vee\neg X$ is an irrational sentence and can be denoted as IR, $X\vee\neg X=IR$. Accordingly, the agent that provides irrational sentences/statements is called an *irrational agent*.

In the classical logic, a conjunction of X with any false statement S, $\mu(S)=F$, is false, $\mu(X\wedge S)=F$. Thus, if the knowledge base (KB) of the domain contains such S then the reasoning in this KB is useless. It is an attempt in any useful KB to make it fully consistent (without contradiction that leads to the false statements in the KB).

However, this is difficult to accomplish due to the inevitable incompleteness of descriptions of a real world object used in KBs. We may have two patients with the same description D in KB, but for one of the patients the statement S(D) is true but for another one S(D) is false, that is one of them has cancer and other one does not. Thus, S(D) is an irrational statement in our terms, we will write this as, S(D)=(T,F), that is assigning to S(D) a vector of values (T, F).

The concept of the IR statement allows us to build a logic of uncertainty that separates *reasoning with IR sentences* from reasoning without them. The reasoning without IR sentences can follow *classical reasoning rules*, while the reasoning with IR can be done differently.

2.4 Conflicting and Irrational Evaluation by a Set of Agents

In the proposed agent approach to the logic of uncertainty, each agent gives logic values to the statements including the irrational evaluations. Thus, we have a *vector of logic evaluations*, e.g.,

$$t(A) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ true & false & true & false \end{bmatrix} \tag{7}$$

In this example, agents together are *incoherent* for statement A because for two agents A is true and for two other agents A is false. We can associate a *vector evaluation t*(*A*) with a numeric *scalar evaluation*, commonly known as a membership value in fuzzy logic

$$\mu(A) = \frac{\mu(Agent_1, A) + ... + \mu(Agent_4, A)}{4} = \frac{1+0+1+0}{4} = \frac{2}{4}$$

Respectively operations $\wedge$ and $\vee$ can be defined as follows

$\mu (X \wedge Y) = \min (\mu (X) , \mu (Y)) - \mu (\neg X \wedge Y) = \mu (Y) - \mu (\neg X \wedge Y)$

$\mu (X \vee Y) = \max (\mu (X) , , \mu (Y)) + \mu (\neg X \wedge Y) = \mu (X) + \mu (\neg X \wedge Y)$

Next, we define a statement S as an *irrational statement* (IR) if *two contradictory values* are given to S by the agent in some order, (TF) or (FT). In addition statements such as $X \wedge \neg X$ =True and $X \vee \neg X$ = false can be called irrational statements, IR.

The agent G is in *irrational state* for statement S if S=IR for the agent G. We also use vector forms (8) and (9).

$$\begin{bmatrix} true \\ false \end{bmatrix}, \begin{bmatrix} false \\ true \end{bmatrix} \tag{8}$$

$$\begin{bmatrix} true \\ true \end{bmatrix} = [true], \begin{bmatrix} false \\ false \end{bmatrix} = [false], \tag{9}$$

In (9) values are repeated therefore to shorten notation we will also use [true] and [false] as show in (9). Form (8) is applicable only to irrational agents (8) and form (9) is applicable both types of agents. Irrational agents can use form (9) saying that $A \vee \neg A$ is always false for these agents,

$$t(\neg A \vee A) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ false & false & false & false \end{bmatrix} \tag{10}$$

In our explanatory model, the irrationality statement (10) is a result of A and ¬A evaluated by each agent in a vector form as irrational statements shown below.

Here every agent first evaluates A as $\begin{bmatrix} false \\ true \end{bmatrix}$ and then evaluates ¬A as $\begin{bmatrix} true \\ false \end{bmatrix}$,

$$t(A) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ \begin{bmatrix} false \\ true \end{bmatrix} & \begin{bmatrix} false \\ true \end{bmatrix} & \begin{bmatrix} false \\ true \end{bmatrix} & \begin{bmatrix} false \\ true \end{bmatrix} \end{bmatrix}$$

$$t(\neg A) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ \begin{bmatrix} true \\ false \end{bmatrix} & \begin{bmatrix} true \\ false \end{bmatrix} & \begin{bmatrix} true \\ false \end{bmatrix} & \begin{bmatrix} true \\ false \end{bmatrix} \end{bmatrix}$$

In this notation, we introduce a *matrix logic* that includes matrix evaluations and matrix operations. Using vector logic evaluations (8) and (9) we can generate a set of matrix evaluations that represent rationality/irrationality of a set of agents for *IR* = X∨¬X as it is shown in the example below for four agents. We remark that four states presented in (8) and (9) are possible for any agent.

Vector and matrix representations of logic were studied in several publications [4], [9], [10], [11]. In [9] a Boolean vector of logic evaluations is assigned to statement S and classical logic operations is applied to each coordinate of the vector evaluation. We expand further this representation to a matrix of Boolean evaluations of S,

$$t(S) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ \begin{bmatrix} s_{11} \\ s_{21} \end{bmatrix} & \begin{bmatrix} s_{12} \\ s_{22} \end{bmatrix} & \begin{bmatrix} s_{13} \\ a_{23} \end{bmatrix} & \begin{bmatrix} s_{14} \\ s_{24} \end{bmatrix} \end{bmatrix}$$

Matrix Boolean operations are defined by applying classical logic operations to each cell individually. In (11) “ ” means any binary logic operation, e.g., ∧,∨, ⊕.

$$t(S \square Q) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ \begin{bmatrix} s_{11} \square q_{11} \\ s_{21} \square q_{21} \end{bmatrix} & \begin{bmatrix} s_{12} \square q_{12} \\ s_{22} \square q_{22} \end{bmatrix} & \begin{bmatrix} s_{13} \square q_{13} \\ a_{23} \square q_{23} \end{bmatrix} & \begin{bmatrix} s_{14} \square q_{14} \\ s_{24} \square q_{24} \end{bmatrix} \end{bmatrix} \tag{11}$$

The negation $t(\neg S)$ is defined by negating each s_{ij} individually as we did in the previous section. This matrix logic can be expanded further from $2 \times n$ matrixes to $m \times n$ matrixes by adding more rows.

3 Elkan's Expressions and Irrationality

First we *irrationalize* each side of ¬(A∧¬B) = B∨(¬A∧¬B) and then compare them showing that irrationalized side are not equivalent. We convert ¬B to ¬B⊕ IR, combine it with *A* producing A∧(¬B⊕IR). The we negate it, ¬(A∧(¬B⊕ IR)) and combine with IR to produce ¬(A∧(¬B⊕ IR)) ⊕IR that is an irrationalized form of ¬(A∧¬B). This process converts rational ¬B to irrational ¬B⊕ IR. The justification for such transformation is provided in section 2.2 with property (6), μ (¬X) = μ (¬X ⊕ IR), known for the classical logic if IR = (S ∧¬S).

3.1 First Expression

Consider statement ¬B evaluated by four agents as shown in (12),

$$t(\neg B) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ true & false & true & true \end{bmatrix} \tag{12}$$

Next we convert ¬*B* to ¬*B*⊕ *IR* using *t*(*IR*) given in (11)

$$t(\neg B \oplus IR) =$$

$$\begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ true \oplus false & false \oplus true & true \oplus false & true \oplus true \end{bmatrix} = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ true & true & true & false \end{bmatrix} \tag{13}$$

This allows us to write *t*(*A*∧(¬*B*⊕ *IR*)) using *t*(*A*) from (7)

$$t(A \wedge (\neg B \oplus IR)) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ true & false & true & false \end{bmatrix}$$

This is a base to produce *t*(¬(*A*∧(¬*B*⊕ *IR*)) ⊕*IR*),

$$t((\neg(A \wedge (\neg B \oplus IR))) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ false & true & false & true \end{bmatrix}$$

$$t(\neg(A \wedge (\neg B \oplus IR)) \oplus IR) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ false & false & false & false \end{bmatrix} \tag{14}$$

3.2 Second Expression

The irrationalized version of *B*∨(¬*A*∧¬*B*) using XOR is

$$B \vee (\neg A \oplus IR) \wedge (\neg B \oplus IR)$$

with the same *A* and *B* that we used in the first expression and using (13) we get

$$t(\neg A) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ false & frue & false & frue \end{bmatrix}$$

$$t(\neg A \oplus IR) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ false & false & false & false \end{bmatrix}$$

$$t((\neg A \oplus IR) \wedge (\neg B \oplus IR)) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ false & false & false & false \end{bmatrix}$$

$$t((B \vee (\neg A \oplus IR) \wedge (\neg B \oplus IR))) = \begin{bmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 \\ false & true & false & false \end{bmatrix} \quad (15)$$

The comparison of (14) and (15) shows that ¬ (A ∧ ¬B) ≠ B ∨ (¬ A ∧ ¬B) when we have irrational agents. It explains why classical equivalence is violated. We view fuzzy logic as a scalar version of this vector logic that uses formula (16).

$$\mu(S) = \frac{1}{n}\sum_{i}^{n} \mu(Agent_i, S) \quad (16)$$

The number of all true values in (14) is 0, but it is 1 in (15). Thus μ (¬(A∧¬B)) = 0, and μ(B∨(¬A¬B) = 1/4 are different for classically equivalent statements. In this way, the presented vector and matrix logic provides an explanatory model for the Elkan's paradox.

4 Conclusion

This paper presented a logic of irrational agents. The semantics of situation that is modeled by this logic is a much more complex that is modeled by the scalar classical logic with its equivalence. We defined vector/matrix representation and logic operations for sets of agents. This provides a new way to study the uncertainty phenomena. The matrix and vector logic approach is a new instrument to study the irrationality and inconsistency of agents. Agents are modeled as a group by using a special type of logic evaluations - matrixes. In this way, we change the individual scalar classical true and false values to a global or vector definition of True and False for a set of agents. The proposed explanatory model resolves the Elkan's paradox. We had shown that for irrational agents the Elkan's paradox is not a paradox but the explained phenomenon. This is applicable to fuzzy logic interpreted as a scalar logic of a mixture of agents that includes self-conflicting and irrational agents. If the agent that uses only T/F evaluations is consistent and rational (has no self-conflict and does not evaluate tautology x∨¬x as false and contradiction x∧¬x as true) the logic equivalence for such agents should be the same as in the classical logic. If such consistent and rational agents use evaluation values from the whole interval [0,1], then we do not see a reason why classical equivalence should be abandoned and broken. This can be a subject of further exploration.

In this paper, we formalized *multiple evaluations* in the vector/matrix logic while the multi-valued logics formalize a *single scalar evaluation* that has three or more values in [0,1] interval in contrast with classical logic.

References

1. Resconi, G., Klir, G., St. Clair, U.: Hierarchical uncertainty metatheory based upon modal logic. Int. J. of General Systems 21, 23–50 (1992)
2. Resconi, G., Türkşen, I.B.: Canonical forms of fuzzy truthoods by meta-theory based upon modal logic. Information Sciences 131, 157–194 (2001)
3. Resconi, G., Kovalerchuk, B.: The Logic of Uncertainty with irrational Agents, JCIS (2006)
4. Resconi, G.: Agents and Lattice, ISKE (2006)
5. Resconi, G., Türkşen, I.B.: Truthoods based on an Additive Semantic Measure with Break of Global Symmetry in model Logic. IJF (2006)
6. Elkan, C.: The Paradoxical Success of Fuzzy Logic. IEEE Expert with fifteen responses 3(8), 9–46 (1994)
7. Elkan, C.: The Paradoxical Controversy over Fuzzy Logic. IEEE Expert , 47–49 (1994)
8. Resconi, G., Jain, L.C.: Intelligent Agents, Theory and Applications. In: Studies in Fuzziness and Soft Computing, vol. 155, Springer, Heidelberg (2004)
9. Westphal, J., Hardy, J.: Logic as a Vector System. Journal of Logic and Computation 15(5), 751–765 (2005)
10. Gaines, B.R.: Fuzzy reasoning and the logics of uncertainty, Multiple-Valued Logic. In: Proceedings of the sixth international symposium on Multiple-valued logic, Logan, Utah, pp. 179–188 (1976)
11. Kovalerchuk, B.: Analysis of Gaines's logic of uncertainty. In: Turksen, I.V. (ed.) A Quarter Century of Fuzziness, Approximate Reasoning and Knowledge Engineering. In: Proc. of NAFIPS 1990 conference, Toronto, vol. 2, pp. 293–295 (1990)
12. Kovalerchuk, B., Resconi, R.: Break of Logic Symmetry by Self-conflicting Agents: Descriptive vs. Prescriptive Rules. In: IEEE Intern. Conference on Integration of Knowledge Intensive Multi-Agent Systems, KIMAS 2007, Boston, pp. 348–353 (2007)
13. Entemann, C.: Fuzzy Logic: Misconceptions and Clarifications. Artificial Intelligence Review 17(1), 65–84 (2002)
14. Pelletier, F.J.: On Some Alleged Misconceptions about Fuzzy Logic. Artificial Intelligence Review 22(1), 71–82 (2004)

Entropy-Based Modeling and Simulation of Evolution in Biological Systems

Stanislaw Sieniutycz

Faculty of Chemical Engineering, Warsaw University of Technology,
00-645 Warsaw, 1 Warynskiego Street, Poland
sieniutycz@ichip.pw.edu.pl

Abstract. We report computer-aided modeling and simulation of evolution in biological systems with living organisms as effect of extremum properties of classical statistical entropy of Gibbs-Boltzmann type or its associates, e.g. Tsallis q-entropy. Evolution for animals with multiple organs is considered. A variational problem searches for the maximum entropy subject to the geometric constraint of constant thermodynamic distance in a non-Euclidean space of independent probabilities p_i, plus possibly other constraints. Tensor dynamics is found. Some developmental processes progress in a relatively undisturbed way, whereas others may terminate rapidly due to inherent instabilities. For processes with variable number of states the extremum principle provides quantitative eveluation of biological development. The results show that a discrete gradient dynamics (governed by the entropy) can be predicted from variational principles for shortest paths and suitable transversality conditions.

1 Introduction

Growing systems (those capable of increasing their size and/or number of states) can exhibit critical behavior when their size increases beyond a certain value. In effect, some developmental processes, i.e. biological evolution, may progress in a relatively undisturbed way, whereas others may terminate rapidly due to strong instabilities. To describe such phenomena quantitatively, we present an extremum principle for entropy S of a physical or a biological system with variable number of states, thus making it possible to investigate processes of biological development and evolution. The extremum principle is of the variational nature and may be formulated as the problem of maximum S subject to the geometric constraint of the constant thermodynamic distance in a (generally non-Euclidean) space of independent probabilities p_i, plus possibly other constraints. The dynamics found are presented in the tensor form. An essential result implies that various dynamics, in particular those of growth processes (characterized by the increase in number of states), are governed by the gradient of the entropy in a Riemannian space.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 34–41, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Role of Complexity and Entropy

In the thermodynamic theory of evolution extrema of complexity Γ with respect to entropy S provide important information. There is a multitude of complexity measures in the literature, all capturing some aspects of what we mean when we say a process is complex. According to Saunders and Ho [1, 2] the complexity growth is the most probable effect in evolving systems. Complexity Γ is a function of disorder D and order Ω, which, in turn, are functions of the information-theoretic entropy S and number of possible states n. One issue is particularly important: a nonequililibrium entropy has to be necessarily applied because the difference between the maximum entropy and the actual system's entropy governs the organization in a complex system. Schrödinger [3] has defined the disorder D and order Ω as expotential functions of non-dimensional quantities S and $-S$ respectively, in units of k_B.Yet, as pointed out by Landsberg these definitions are inappropriate for growing systems [4]. For such systems, Landsberg's definitions of disorder $D$and order Ω are used

$$D = S/S_{\max} \tag{1}$$

$$\Omega = 1 - D = 1 - S/S_{\max} \tag{2}$$

As S_{max} depends on the number of states, n, in Landsberg's definition both disorder D and order W are functions of the information entropy S and number of states, n, i.e. and . One especially simple form is the complexity

$$\Gamma_n(S) \equiv 4D\Omega = 4(S/S_{\max(n)})(1 - S/S_{\max(n)}). \tag{3}$$

The coefficient 4 in Eq. (3) is introduced to normalize the quantity Γ_n. The subscript n refers to a complexity sequence in a system with growing number of states, n.

The solution to the following equation

$$d\Gamma_n(S)/dS = 0 \tag{4}$$

allows one to determine the extremum value of the information-theoretic entropy, $S^{\wedge}$, which maximizes complexity Γ_n [5]. The maximum attained by the function $\Gamma_n = \Gamma(S_n)$ equals the unity. This maximum appears for $D = 0.5$. Hence the entropy maximizing the complexity in this model equals to one half of the maximum entropy (achieved at the total randomness)

$$S^{\wedge} = \frac{1}{2} S_{\max}(n). \tag{5}$$

Evolution occurs on submanifolds that are surfaces of constant entropy [1, 5]. Here the generalized (Tsallis) entropy $S$is used in terms of probabilities p_0, p_1, ...p_n

$$S = (q-1)^{-1}\left(1 - \sum_{k=1}^{n} p_k^q\right) \tag{6}$$

where p_k is the probability of finding an element in the state $i$among n states possible and $\sum p_k$=1. Performing maximization of S one can easily show that for a sole constraint $\sum p_k$=1 maximum of entropy occurs for the total randomness. All probabilities satisfy then the equality p_i=p_k =$n^{-1}$and the maximum entropy is

$$S_{\max} = (q-1)^{-1}\left(1 - n^{1-q}\right). \tag{7}$$

In the classical case ($q \to 1$) this formula yields a well-known result, S_{max}= ln$n$.

Yet, in an example considered, a multi-organ animal is a system with $2n+1$ probabilities. They describe $n$pair of legs plus a remaining part of the body, hence

$$S_{\max} = (q-1)^{-1}\left(1 - (2n+1)^{1-q}\right). \tag{8}$$

In the classical case ($q \to 1$) this formula yields, S_{max}= ln$(2n+1)$. The complexity-maximizing entropy equals to one half of these quantities. In the classical case of q=1 one obtains

$$S_n^{\wedge} = (\frac{1}{2})\ln(1+2n) \tag{9}$$

Generally, the complexity-maximizing entropy is

$$S_n^{\wedge} = (\frac{1}{2})S_{\max} = \frac{1}{2}(q-1)^{-1}[1-(1+2n)^{1-q}], \tag{10}$$

where of Eq. (9) follows in the classical case of q=1. In order to work with independent probabilities we eliminate the last probability from the entropy expression (6) with the help of normalization condition $\sum p_i$=1. We then obtain a tilde entropy function

$$S = S^{\sim}(p_0, p_1, ..., p_{n-1}) \tag{11}$$

and, from Eq. (3), a related complexity

$$\Gamma_n^{\sim} = F_n[S^{\sim}(p_0, p_1, ..., p_{n-1})]. \tag{12}$$

Using independent probabilities we work with a tilde entropy function, $S^{\sim}$. For the evolutions satisfying the maximum complexity principle [1], the entropy equals to the complexity-maximizing entropy , consistent with the statement of Saunders and Ho [1] "The only completely reversible changes are those which are isocomplex". In fact, when $(p_0, p_1, ..p_{n-1})$, a subset of probabilities $p = (p_0, p_1, ..p_{n-1})$is found describing the evolution submanifolds. The evolution submanifolds are those described by the equality

$$S^{\sim}(p_0, p_1, ..., p_{n-1}) = S_n^{\wedge} \tag{13}$$

and assuring the value of $S_n^{\wedge}=(1/2)S_{max(n)}$. Within this manifold, a reversible modification (specialization, mutation) of states is possible. Thus, in the evolution examples, the solutions to equality$S(p_0, p_1, ..p_{n-1}) = (1/2)S_{max(n)}$ refer to the *submanifolds of evolution*, or surfaces on which modifications (mutations) of organs may occur.

3 An Organism with Multiple Organs Without Mutations

Following earlier works along this line [1, 2, 5, 6] we analyse here the evolution of a multiple-organ organism, e.g. trilobite, an animal with many pair of legs. Although trilobites died out millions years ago, their anatomical structure is known from excavations. For our purposes it is enough to distinguish one pair of legs, of probability$2p_i$ from the remaining parts of the organism that has probability p_0. For a multi-organ animal with n pairs of legs

$$S = (q-1)^{-1}\left(1 - p_0^q - 2np_1^{1-q}\right). \tag{14}$$

and, in the classical case,

$$S = -p_0 \ln p_0 - 2np_1 \ln p_1. \tag{15}$$

These formulae both hold subject to the condition of $\sum p_i$=1, from which

$$p_1 = 1 - p_0/2n. \tag{16}$$

Whence the generalized entropy in terms of single independent probability

$$S^{\sim}(p_0) = (q-1)^{-1}\left(1 - p_0^q - (2n)^{1-q}(1-p_0)^q\right). \tag{17}$$

and its classical limit of $q = 1$

$$S^{\sim}(p_0) = -p_0 \ln p_0 - (1-p_0)\ln[(1-p_0/2n]. \tag{18}$$

Each of these entropies must be equal to its complexity-maximizing counterpart as described by Eq. (10) and its classical limit (9). By comparing Eq. (10) and (17), that hold valid for an arbitrary q, we obtain equations describing independent probabilities in terms of n (sole p_0 in our example)

$$(q-1)^{-1}\left(1 - p_0^q - (2n)^{1-q}(1-p_0)^q\right) = \frac{1}{2}(q-1)^{-1}[1-(1+2n)^{1-q}] \tag{19}$$

and in the case of classical entropy

$$-p_0 \ln p_0 - (1-p_0)\ln[(1-p_0/2n] = (\frac{1}{2})\ln(1+2n). \tag{20}$$

From these equations probabilities p_0 and (then)p_1 can be calculated in terms of n. Graphs of these results for $q = 1$ are presented in refs. [5] and [6], where appropriate results are restricted to points describing the evolution without modifications or specializations.

4 Mutations and Specializations of Organs

Here we analyze the evolution of a multiple-organ animal with mutations or specializations. An example relevant to this case follows the scheme including the stage trilobite $\longrightarrow$crab.

With entropies S expressed in terms of independent probabilities p_i (functions $(p_0, ...p_{n-1})$ one may consider effects of reversible modifications (mutations) of multiple organs (e.g. pair of legs), for a fixed value of S. In the considered example, after modification of a pair of legs to claws a crab emerges from a trilobite. Considering the anatomical structure of the crab, one pair of claws is distinguished with probability $2p_2$.

When the modification occurs without change in number of pairs of legs and claws, the following equality holds

$$2p_2 + 2(n-1)p_1 + p_0 = 1. \tag{21}$$

For an organism with one pair of organs modified (specialized) on the reversibility surface, Eq. (13) and (21) are applied in the space of independent probabilities to describe the equality of the generalized entropy $S^{\sim}$ and the complexity-maximizing $S_n^{\wedge}$. We obtain

$$(q-1)^{-1}\,(1-p_0^q-(2n-1)p_1^q-2p_2^q(p_0,p_1)) = \frac{1}{2}(q-1)^{-1}[1-(1+2n)^{1-q}] \tag{22}$$

in the generalized case, and

$$-p_0 \ln p_0 - 2(n-1)p_1 \ln p_1 - 2p_2(p_0,p_1)\ln[p_2(p_0,p_1)] = (\frac{1}{2})\ln(1+2n) \tag{23}$$

in the classical case. The probability function $p_2(p_1, p_0)$used in these equations has the form

$$p_2(p_0,p_1) = \frac{1}{2}(1-p_0-2(n-1)p_1) \tag{24}$$

that follows from the condition $\sum p_i = 1$represented by Eq. (21). Note that$2p_2$is the probability attributed to the modified organ. The complexity-maximizing

entropies ($S_n^{\wedge}$=$S_{max}/2$) are those used earlier, Eq (10). The evolution submanifolds are now the family of lines $p_0(p_1, n)$. They describe organisms possesing $n-1$ of identical organs (pairs of legs) and one organ being modified, specialized, or subjected mutations. A special subset of data refers to organisms without specialization (Sec.3).

In the evolution literature Williston's law is frequently quoted [1, 2], which subsumes the results of observation and comparative analysis. This law states that if an organism possesses many of the same or similar elements, a tendency appears to reduce the number of these elements along with the simultaneous modification (specialization) of these elements which are saved by the organism. In the example, the evolution submanifolds describe organisms with $n-1$ of identical organs (pairs of legs) and one organ being modified, specialized, or subjected mutation. Spontaneous increase of complexity is here a basic feature of evolution. For biological systems a reasonable measure of complexity is the different number of components they contain, so as to be consistent with well-known Williston's law which predicts that very similar components will either merge or specialize. With the idea of increasing complexity, the system's organization acts as a force which prevents loss of components and allows complexity increase by the integration of new ones. This leads to a principle stating that an organism with more of organs is more susceptible to evolution towards an increase in the number of these organs. Yet, during reversible specialization of organs, the state of an organism can fall into the region of the catastrophic decrease of number of these organs. These catastrophes constitute the price of specialization. The likelihood of falling in the catastrophe region increases with the number of organs. This explains why organisms possessing large number of similar organs ultimately reduce this number, despite the fact that they are more susceptible to evolutionary increase in the organ number. This also agrees with the well-known Williston's law of evolution [1] that subsumes the results of observation and is confirmed by the excavation experiments. In the dynamical description of this problem an extremum principle provides a quantitative picture of biological development. It shows that a discrete gradient dynamics (governed by the entropy potential) can be predicted from variational principles for shortest paths and suitable transversality conditions.

5 Variational Dynamics of Evolution

Working in the dynamical context [5] we may analyze the evolution of living organisms treated as multi-organ systems by using the complexity criterion based on a potential, usually taken as the classical statistical entropy (of Shannon-Boltzmann) and the entropy-based complexity. Here, however, in order to penetrate a vaster spectrum of stability (instability) properties, the generalized Tsallis entropy S is used (in k_B units) as a function of independent probabilities p_0, p_1, ...p_n. In the analysis of this sort, classical thermodynamic quantities do not appear, yet the model used satisfies an extremum principle that, similarly as in thermodynamics, implies the maximum of entropy subject to the geometric

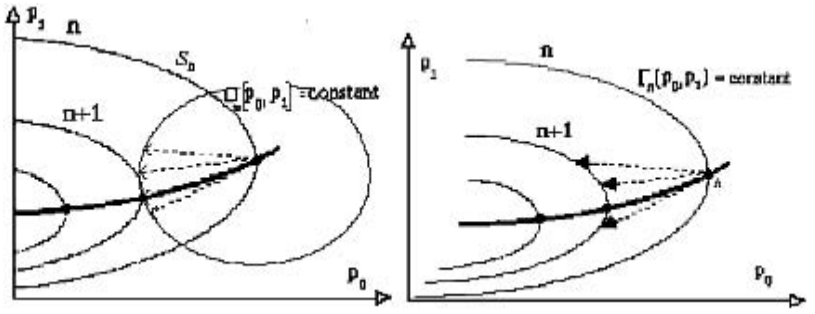


Fig. 1. An original optimization problem for a system with$n$ states is that of maximum change of entropy or complexity Γ between a point and a surface of a constant distance from the point. Dual optimization problem is that of minimum length between a point and a surface of constant entropy for the system with $n + 1$states.

constraint of a given thermodynamic distance. Specifically, an original optimization problem for a system with$n$ states is that of maximum change of entropy or entropy-related complexity Γ between a point and the surface of constant distance from the point (Fig.1, left). Dual forms of this principle can also be stated, where one mimimizes the thermodynamic length subject to a fixed change of the system's complexity or entropy (Fig.1, right).

For practical purposes the dual problem is analyzed which is the problem of minimum length between a point (for a system with n states) and the entropy manifold of the system with n+1 states. In this formulation, an s-dimensional evolution can be obtained via minimization of the length functional

$$J = \int_{p_0^n}^{p_0^{n+1}} a(p_0, p_1)\sqrt{1 + (dp_1/dp_0)^2}dp_0, \tag{25}$$

where the independent probabilities $p_i$are constrained to reside on the constant-entropy manifold satisfying the constraint $dS^{\sim}$=0. It may then be shown [5] that both the original problem of maximum entropy and its dual problem of the minimum length yield an evolution dynamics in the gradient form. This form holds also in curvilinear spaces. In fact, admitting non-flat metrices (i.e. the situation when a Lagrangian associated with a non-flat metric is effective) we have shown that the tensor equation of the continuous model has the form

$$\frac{dp_i}{d\tau} = g_{ik}\frac{\partial S^{\sim}(p_0, ...p_s)}{\partial p_i}, \tag{26}$$

where $\tau = \omega t$ is a nondimesional time and ω is the reciprocal of a time constant.

The consequence of this equation is the tensor form of the discrete evolution

dynamics with the Onsager-like structure, where his symmetry matrix $L_{ik} = \Delta t \omega g_{ik}$ appears

$$p_s(n+1) - p_s(n) = L_{s1} \frac{\partial S^{\sim}_{n+1}(p_0, ...p_s)}{\partial p_0} + ... + L_{ss} \frac{\partial S^{\sim}_{n+1}(p_0, ...p_s)}{\partial p_s}, \quad (27)$$

as in classical irreversible thermodynamics. Therefore, the evolution process can be imbedded into a relatively large family of thermodynamic processes.

In conclusion, by applying the tensor calculus, one can thus develop a discrete, nonlinear generalization of evolution dynamics in metric spaces that may be curvilinear. Dynamic programming algorithms (Bellman's equations) can be derived and computer-aided simulations of their solutions can be performed. Systems governed by nonclassical q-entropies may exhibit quantitatively distinct picture of instabilities than classical. Evolutions of living organisms can be described by variational principles for maximum of entropy along with suitable transversality conditions. Gradient dynamics that governs the evolution problems is of Onsager's structure and is consistent with the entropy principle of extremality as the driving factor in the discrete dynamics of complex and living systems, postulated recently [5]. We have shown that such a principle should be integrated with the evolution theory of biological systems.

Acknowledgement. The author acknowledges a support from the KBN, grant T09C 024 26: Non-equilibrium thermodynamics and optimization of chemical reactions in physical and biological systems.

References

1. Saunders, P.T., Ho, M.W.: On the increase in complexity in evolution. I. J. Theor. Biol. 63, 375–384 (1976)
2. Saunders, P.T., Ho, M.W.: On the increase in complexity in evolution II. The relativity of complexity and principle of minimum increase, J. Theor. Biol. 90, 515–530 (1981)
3. Schrodinger, E.: What is life? Cambridge University Press, Cambridge (1967)
4. Landsberg, P.T.: Can entropy and order increase together. Phys. Lett. 102 A, 171 (1984)
5. Szwast, Z., Sieniutycz, S., Shiner, J.: Complexity principle of extremality in evolution of living organisms by information-theoretic entropy. Chaos, Solitons and Fractals 13, 1871–1888 (2002)
6. Szwast, Z.: An approach to the evolution of selected living organisms by information-theoretic entropy (in Polish), Reports of Faculty of Chemical and Process Engineering at Warsaw TU 24, 123–143 (1997)

Logistic Regression as a Computational Tool for Dealing with Intransitivity

María Isabel Rodríguez-Galiano and Jacinto González-Pachón

Artificial Intelligence Department,
School of Computer Science, Universidad Politécnica de Madrid
{irodriguez,jgpachon}@fi.upm.es

Abstract. In this paper, we propose a decision-making methodology based on logistic regression to solve a general decision-making problem, considering imprecisions and incoherences in the decision maker's behaviour.

Keywords: Paired comparisons, intransitivities, multinomial logistic regression, ranking.

1 Introduction

Some degree of inconsistency is inherent to human decision making. Inconsistency in discrete choices may be due to the fact that a large number of alternatives have to be taken into account. One problem for preference modelling then is that decision makers' (DMs) judgements may not be transitive, see [1], [2] and [3] for example.

From a normative point of view, intransitivity is a problem because transitivity is a fundamental principle for studying a choice process as an optimisation problem. However, people exhibit intransitivity and, sometimes, do not want to change their minds. In this paper, we propose a methodology for dealing with intransitivity without changes having to be made.

May [1] attributed intransitivity to the multiple components involved in the decision-making process and he proposed analysing preference patterns in transitive components. Taking up this idea, we propose decomposing an original binary relation (representing DMs preferences) into transitive binary relations as a first phase of our methodology. The decomposition phase will be based on the Mixture of Maximal Quasi Order (MMQO) notion published in González-Pachón and Ríos-Insua [4].

Each transitivity component may be interpreted as implicit criteria arising in the DMs mind during the pairwise comparison process [5]. Therefore, we propose aggregating these transitive components in a second phase of our methodology. For computational reasons, the logit model has proven to be an important aggregation tool for dealing with discrete choice models [6], [7]. In this paper, we propose a multinomial logistic regression model as a computational tool for dealing with inconsistency in a discrete choice process.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 42–49, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The paper is organised as follows. The decomposition phase is stated in Section 2. The aggregation phase is described in Section 3. We then use numerical examples to illustrate the above ideas in Section 4. Finally, we present the main conclusions derived from this research.

2 Decomposition Phase

To deal with intransitivities, the first step is to decompose a binary relation into transitive components.

Let R be a reflexive binary relation over the set of alternatives $\Omega = \{x_1, \ldots, x_n\}$. Let $V = \{R_i\}_{i \in I}$ be the set of all quasi orders (i.e., reflexive and transitive binary relations) included in R. If we consider the inclusion as an order, V may be considered as an ordered set. We say that $R^* \in V$ is a maximal quasi order included in R if there is no $R' \in V$ such that $R^* \subset R'$.

The family of all maximal quasi orders included in a reflexive binary relation R will be called the mixture of maximal quasi order (MMQO) associated to a binary relation R and will be denoted by $\mathcal{F}$. Obviously, $\mathcal{F}$ is included in V.

The following representation result was stated in [4].

Theorem 1. *If R is a reflexive binary relation on a finite set Ω, the associated MMQO, $\mathcal{F}$, is not empty and covers R, i.e.*

$$\mathcal{F} \neq \emptyset \quad and \quad R = \bigcup_{Q \in \mathcal{F}} Q$$

An algorithm for searching the family $\mathcal{F}$ is given in [8].

Conflicts between quasi orders contained in $\mathcal{F}$ can explain the intransitivities that appear in the original DM's preferences.

A later phase can be defined to aggregate the transitive components.

3 Aggregation Phase

In the aggregation phase we used a logistic regression model to get a valued binary relation from the family of maximal quasi orders associated with a reflexive binary relation R. Our final model is summarized in the flow chart shown in Figure 1.

We used a generalization of logistic regression models that handle multicategory or multinomial responses and several explanatory variables.

The Multinomial Logistic Regression Model

Let $\mathcal{F} = \{R_i\}_{i=1}^{k}$ be the MMQO associated with R. Let us define the variables on which to apply logistic regression. Independent variables will express binary choice sets, while dependent variable will represent the alternative selected by the DM. We denote the predictors for a nominal response Y as $X_1, \ldots, X_n$. If we consider the subset $\{x, y\} \subset \Omega$, the predictor variables are defined as follows.

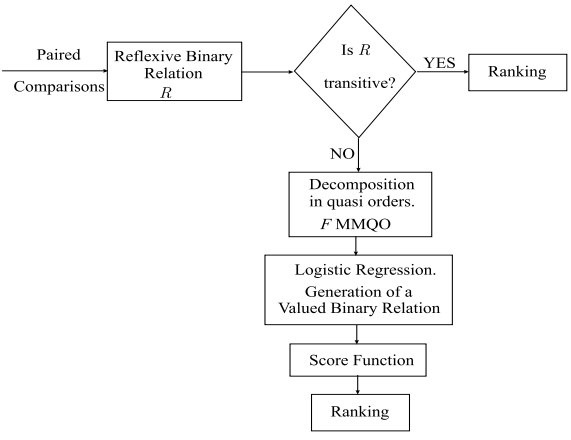


Fig. 1. Our model

$$X_i = \begin{cases} 1 & \text{if } x_i \in \{x,\ y\} \\ 0 & \text{if } x_i \notin \{x,\ y\} \end{cases} \ i = 1, \ldots,\ n$$

The response variable Y represents the preference structure associated with R, that is, the triplet of relations $\{P,\ I,\ N\}$, representing strict preference, indifference and incomparability relations, respectively [9]. Given a subset $\{x_s,\ x_t\}$ where $s \neq t$ and $s, t \in \{1, \ldots,\ n\}$,

$$\begin{aligned} Y = s &\ \text{ iff } \ x_s P x_t \\ Y = t &\ \text{ iff } \ x_t P x_s \\ Y = 0 &\ \text{ iff } \ x_s N x_t \end{aligned}$$

Therefore, the preference relation is represented by the number of the preferred alternative and the incomparability relation by 0, indicating that the choice within the corresponding subset is empty.

If the alternatives are indifferent, the selection subset will include both alternatives, and then an observation has to be added to each alternative to represent the indifference relation. Hence, if $x_s I x_t$, the observations are

$$\begin{aligned} (X_s = 1, X_t = 1, X_j = 0\ \forall j \neq s, t),\ \ Y = s \\ (X_s = 1, X_t = 1, X_j = 0\ \forall j \neq s, t),\ \ Y = t \end{aligned}$$

The multinomial logistic regression model provides an estimation of response probabilities $\{\pi_i\}_{i=0}^{n}$, i.e. $\pi_i = P(Y = i)\ \ i = 0, \ldots, n$, where $\sum_i \pi_i = 1$, as a function of variables $X_1, \ldots,\ X_n$. The function over variables $X_1, \ldots,\ X_n$ can be a linear function of the observations and/or have non-linear terms (interactions between variables). We use a linear function. Therefore, the response probabilities are defined in this model as

$$\pi_i = \frac{exp(\alpha_i + \beta_{1i}X_1 + \ldots + \beta_{ni}X_n)}{\sum_{h=0}^{n} exp(\alpha_h + \beta_{1h}X_1 + \ldots + \beta_{nh}X_n)}$$

for each $i = 0, \ldots, n-1$. See [10] for more details about these models.

Let $\pi_0^{x,y}$, $\pi_1^{x,y}$, ..., $\pi_n^{x,y}$ be the response probabilities for each subset of alternatives $\{x,\ y\}$, i.e., $\pi_i^{x,y}$ is the probability that the response Y falls into category i for observation $\{x,\ y\}$. Thus, if $x = x_s$ and $y = x_t$ with $s \neq t$ and $s, t \in \{1, \ldots,\ n\}$,

$$\begin{aligned}\pi_i^{x,y} &= P(Y = i|x, y) = \\ &= P(Y = i|X_s = 1, X_t = 1, X_j = 0\ \forall j \neq s, t)\end{aligned}$$

Evaluating Model Fit

As a residual analysis is not appropriate here and diagnostic procedures are not completely formalized [11] [12], we deal with the problem of evaluating model fit as follows.

We can use the correct classification rate as a measure of the model fit.

There is no ready equivalent to the standard regression model's coefficient of determination R^2. A number of alternative similar measures for R^2, called Pseudo-R^2, have been proposed, but none are widely accepted. [13] studied five of them, pointing out that there is no consensus on how those measures should be calculated and used. Most Pseudo-R^2 cannot have a value of 1 even when the model fits the data perfectly, that is, some of these measures do not approach one as was theoretically expected.

Finally, we can also use hypothesis tests to validate whether the model adequately fits the data. The likelihood-ratio test compares the maximized log likelihood (L_1) for the model to the maximized log likelihood (L_0) for the simpler model in which all the parameter coefficients are zero (null hypothesis). The likelihood-ratio test statistic is equal to $-2(L_0 - L_1)$. It has a Chi-squared distribution with degrees of freedom equal to the difference between the residual degrees of freedom values for the two models, if the null hypothesis is true. Two goodness-of-fit tests are based on Pearson and Deviance statistics, they also have a Chi-squared distribution if the null hypothesis is true.

The Valued Binary Relation

From response probabilities we define the following valued binary relation $T : \Omega \times \Omega \longrightarrow [0,\ 1]$ associated with the family $\mathcal{F}$.

$$T(x_s,\ x_t) = \pi_s^{x_s, x_t},\ \forall s, t \in \{1, \ldots,\ n\},\ s \neq t$$

Furthermore, $T(x,\ x) = 1\ \forall x \in \Omega$, i.e. T is reflexive.

The Ranking Method

Now, we have to associate a score based on T, $\mathcal{S}(x,\, T)$, with each alternative $x \in \Omega$ and to rank alternatives according to these scores.

We consider the valued net flow rule, that associates a weak order (a reflexive, transitive and complete relation) to each reflexive valued relation [14]. It is the net flow rule applied to valued relations that rank orders the alternatives according to the following score:

$$\mathcal{S}(x,\, T) = \sum_{y \in \Omega \setminus \{x\}} (T(x,\, y) - T(y,\, x))$$

Thus, a ranking method based on $\mathcal{S}(x,\, T)$ is $\succeq_T$ defined by

$$x_s \succeq_T x_t \iff \mathcal{S}(x_s,\, T) \geq \mathcal{S}(x_t,\, T)$$

$\succ_T$ and $\sim_T$ denote the preference and indifference relations associated with $\succeq_T$, respectively.

4 An Example

Suppose that the DM preferences over the set of alternatives $\Omega = \{x_1, x_2, x_3, x_4\}$ are given by the binary relation R, defined by the graph $G(R)$, in Figure 2.

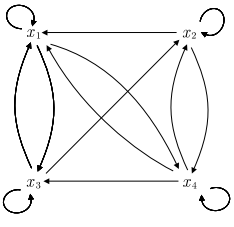


Fig. 2. $G(R)$

The MMQO, $\mathcal{F}$, associated with R is shown in Figure 3.

For example, the preference structure associated to the second relation of family $\mathcal{F}$, R_2, is as follows

$$\begin{aligned} P &= \{(x_2,\, x_1),\, (x_3,\, x_1),\, (x_4,\, x_1)\} \\ I &= \{(x_2,\, x_4)\} \\ N &= \{(x_2,\, x_3),\, (x_3,\, x_4)\} \end{aligned}$$

which yields the data shown in Table 1.

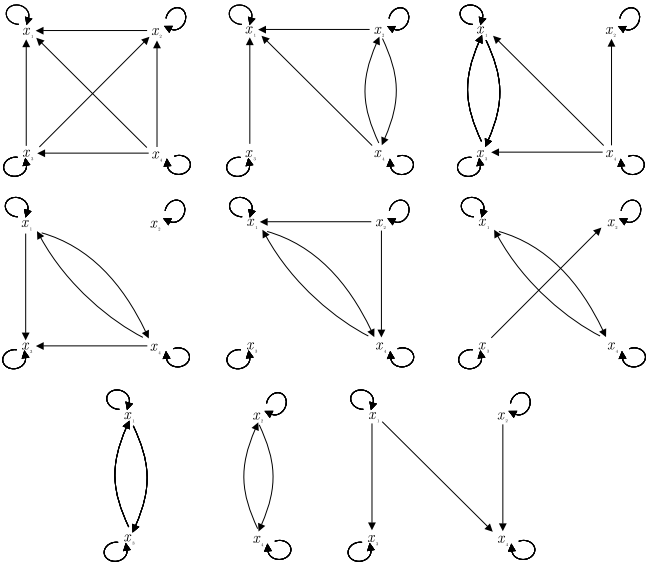


Fig. 3. $\mathcal{F}$

Table 1. Input data corresponding to R_2

x, y	X_1	X_2	X_3	X_4	Y
$x_1,\ x_2$	1	1	0	0	2
$x_1,\ x_3$	1	0	1	0	3
$x_1,\ x_4$	1	0	0	1	4
$x_2,\ x_3$	0	1	1	0	0
$x_2,\ x_4$	0	1	0	1	2
$x_2,\ x_4$	0	1	0	1	4
$x_3,\ x_4$	0	0	1	1	0

We use the estimated response probabilities at each subset of alternatives in the multinomial logistic regression to construct the valued binary relation. These probabilities, generated by the statistical software SPSS 14, are shown in Table 2.

SPSS output gives three Pseudo-R^2 measures: Cox and Snell, Nagelkerke and McFadden. In this example, they are .728, .767 and .437, respectively. They are good taking into account that they are not completely standardized. The likelihood-ratio statistic equals $-2(L_0 - L_1) = 71.707$ with $df = 12$, which yields a p-value of $p < .0001$. Since this p-value is very small we have evidence enough to reject the null hypothesis, and the final model outperform the null model. Both goodness-of-fit tests (Pearson and Deviance) have a chi-squared statistic that equals .001 with $df = 8$ and $p = .998$. This p-value is very high so there is no evidence to reject our model and then it provides a good fit to the data.

Table 2. Estimated response probabilities

x, y	$\pi_0^{x,y}$	$\pi_1^{x,y}$	$\pi_2^{x,y}$	$\pi_3^{x,y}$	$\pi_4^{x,y}$
$x_1,\ x_2$	0.625	0	0.375	0	0
$x_1,\ x_3$	0.2	0.4	0	0.4	0
$x_1,\ x_4$	0.091	0.364	0	0	0.545
$x_2,\ x_3$	0.75	0	0	0.25	0
$x_2,\ x_4$	0.2	0	0.4	0	0.4
$x_3,\ x_4$	0.625	0	0	0	0.375

The valued binary relation T associated to $\mathcal{F}$ is defined as shown Table 3.

Table 3. Valued Binary Relation

$(x,\ y)$	(x_1, x_2)	(x_1, x_3)	(x_1, x_4)	(x_2, x_1)	(x_2, x_3)	(x_2, x_4)
$T(x, y)$	0	o.4	0.364	0.375	0	0.4
$(x,\ y)$	(x_3, x_1)	(x_3, x_2)	(x_3, x_4)	(x_4, x_1)	(x_4, x_2)	(x_4, x_3)
$T(x, y)$	0.4	0.25	0	0.545	0.4	0.375

Table 4 includes the score function.

Table 4. Score Function

x	x_1	x_2	x_3	x_4
$\mathcal{S}(x,\ T)$	-0.556	0.125	-0.125	0.556

Finally, we obtained the following ranking based on $\mathcal{S}(x,\ T)$.

$$x_4 \succ x_2 \succ x_3 \succ x_1$$

5 Conclusions

The objective of this paper was to solve imprecisions and incoherences in the DM's behaviour in pairwise comparisons. We have achieved a ranking by decomposing a binary relation and using the multinomial logistic model. Being implemented by computer tools, this logistic model is an important instrument.

The methodology presented here could be applied to deal with group decision making and to aggregate the binary relations generated by the problem. In this case, we would have a larger sample and the logit model is better suited than in the individual case.

Another interesting context for extending this methodology is the field of choice functions. We can use the results presented in [15] to apply this method to a problem with choice functions, which are another way of describing DM's preference.

Acknowledgments. The work of Mª Isabel Rodríguez-Galiano and Jacinto González-Pachón was partially supported by the Spanish Ministry of Science and Technology under the projects DPI2006-15346-C03-02 and SEJ2005-04392/ECON, respectively. Thanks are due to Mrs. Rachel Elliot for reviewing the English.

References

1. May, K.O.: Intransitivity, Utility, and the Aggregation of Preference Patterns. Econometrica 22(1), 1–13 (1954)
2. Roberts, F.S: What if Utility Functions do Not Exist? Theory and Decision 3, 126–139 (1972)
3. Tversky, A.: Intransitivity of Preferences. Psychological Review 76, 31–48 (1969)
4. González-Pachón, J., Ríos-Insua, S.: Mixture of Maximal Quasi Orders: A New Approach to Preference Modelling. Theory and Decision 47, 73–88 (1999)
5. González-Pachón, J., Rodríguez-Galiano, M.I., Romero, C.: Transitive Aproximation to Pairwise Comparison Matrices by using Interval Goal Programming. Journal of the Operational Research Society 54, 532–538 (2003)
6. Aldrich, J.H., Nelson, F.D.: Linear Probability, Logit, and Probit Models. Sage Publications, Thousand Oaks (1989)
7. Bierlaire, M.: Discrete Choice Models. In: Labbé, M., Laporte, G., Tanczos, K. (eds.) Operations Research and Decision Aid Methodologies in Traffic and Transportation Management. NATO ASI Series, Series F: Computer and Systems Science, vol. 166, Springer, Heidelberg (1998)
8. González-Pachón, J., Romero, C.: A method for obtaining transitive approximations of a binary relation. Annals of Operations Research (to appear, 2007)
9. Roubens, M., Vincke, P.h.: Preference Modelling. Lecture Notes in Economics and Mathematical Systems, vol. 250. Springer, Heidelberg (1985)
10. Agresti, A.: Categorical Data Analysis. Wiley, Chichester (2002)
11. Christensen, R.: Log-Linear Models and Logistic Regression, 2nd edn. Springer, New York (1997)
12. Ryan, T.P.: Modern Regression Methods. Wiley, New York (1997)
13. Menard, S.: Coefficients of Determination for Multiple Logistic Regression Analysis. The American Statistician 51(1), 17–24 (2000)
14. Bouyssou, D.: Ranking Methods based on Valued Preferences Relations: A Characterization of the Net Flow Method. European Journal of Operational Research 60, 61–67 (1992)
15. Rodríguez-Galiano, M.I., González-Pachón, J.: Characterization of Certain Orders using their Associated Choice Functions. European Journal of Operational Research 132, 619–627 (2001)

Synoptic Maps Forecast Using Spatio-temporal Models

J.L. Crespo[1], P. Bernardos[1], M.E. Zorrilla[2], and E. Mora[1]

[1] Department of Applied Mathematics and Computer Sciences, University of Cantabria. Avda. de los Castros s/n 39005 Santander, Spain
{crespoj, bernardp, morae}@unican.es
[2] Department of Mathematics, Statistics and Computation, University of Cantabria. Avda. de los Castros s/n 39005 Santander, Spain
zorrillm@unican.es

Abstract. The objective of this paper is to study several approaches to forecasting the temporal evolution of meteorological synoptic maps that carry information in visual form but without objects. Window-based descriptors are used in order to accomplish continuity so the prediction task is possible. Linear and non-linear models are applied for the prediction task, the first one being based on a spatio-temporal autoregressive (STAR) model whereas the second one is based on artificial neural networks. The method and obtained results are discussed.

1 Introduction

The study of meteorological phenomena is an important problem that requires approximations in different fields. Many advances have been made in the numerical modelization field but few works exist in the image processing framework ([1], [5-8]) which is a reasonable approach when the information is basically of visual nature, as occurred in meteorological synoptic maps.

An image prediction problem is studied here, where optical flow techniques are not applicable because of the special characteristics of the images used.

In our case, synoptic maps show complicated behaviours where high and low pressure areas move, change their shape and even appear or disappear from one map to the next, see, for example, the time evolution shown in Figs. 1 and 2, separated 24 hours.

This is the type of images treated in the following, where low and high pressure centers and isobar lines distribution are drawn. But pressure is not the only information drawn in these images. Details of extracting the relevant information and defining numerical descriptors can be reviewed in previous publications (see [2] and [3]). Nevertheless, their main lines are reviewed here for the sake of completeness.

Numerical descriptors are defined for adequate small windows that the original images are split into. These descriptors are the isobar density and their orientation for the isobar lines, and the positions of H and L letters for the centers of high and low pressure areas. Then, a window by window prediction is carried out for each numerical descriptor separately. The independent variables of the prediction function are the corresponding descriptor values of the window to predict and its spatial neighbours in previous time steps.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 50 – 56, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Two types of dependence for a window with its neighbours are studied: linear and non-linear. The first one is based on a spatio-temporal autoregressive (STAR) model and the second one is based on artificial neural networks, in our case, with classical multilayer perceptron structure.

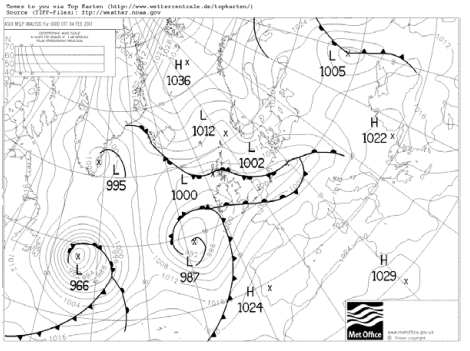


Fig. 1. Synoptic map corresponding to 2001 4th February. Isobar lines as well as high (H) and low (L) pressure centers are drawn

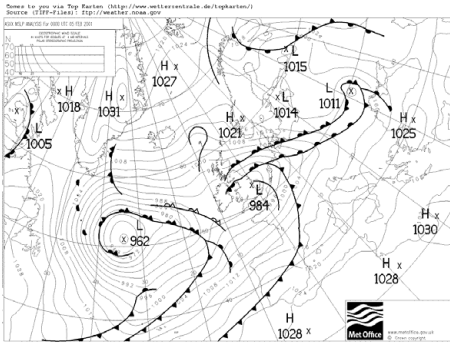


Fig. 2. Synoptic map corresponding to 2001 5th February

2 Linear STAR-Based Model

The linear STAR model is based on the approach made by Pfeifer and Deutsch in the 1980s [9]. It is a three dimensional version of the regular autoregressive (AR) model. While the AR model studies the time evolution of a single point, STAR model considers the time evolution of a spatial grid of points.

STAR models have been applied to different areas such as imaging, hydrology or business. More recently, an image prediction problem focused on images with no identifiable objects is addressed within a STAR-based model [4].

Considering an image divided into small windows, descriptors of a *w* window in a given *t* time can be modeled as a linear combination of descriptors of neighboring windows in space and time as:

$$d_w(t) = \sum_{i=1}^{T} \left[(A)^i d_{w_{neigh}}(t-i) \right] + \alpha(w,t), \tag{1}$$

where $(A)^i$ is the array of autoregressive parameters for each instant of time *i* considered, d_{Wneigh} are the descriptors in the space-neighborhood windows around *w* and α is a noise term, which is assumed to be Gaussian for parameter estimation purposes.

Then, the prediction procedure is divided into two stages: first, a fitting stage where the parameters' matrix is obtained and second, the actual prediction stage.

The parameters of the model are fitted by minimizing the average square error (ASQE):

$$\text{ASQE}=\overline{\sum_{w}(d_w-\hat{d}_w)^2}\,, \tag{2}$$

where d_w is the descriptor value of a window, $\hat{d}_w$ is the descriptor value predicted for the same window and the sum is made to every window of the image. Finally, one obtains the $(A)^{i,min}$, which will be used for the prediction stage as:

$$\hat{d}_w(t)=\sum_{i=1}^{T}\left[(A)^{i,min}\,d_{w_{neigh}}(t-i)\right]. \tag{3}$$

3 Neural Network Model

Artificial neural networks are capable of solving problems of functional approximation. The main difficulty is obtaining a suitable network design. The most popular, the multilayer perceptron, has been used for time series prediction in different fields, and more recently, in the image prediction field [4].

A perceptron with two hidden layers and ordinary logistic activation function is used to examine the relevance of non-linear effects for our descriptor prediction task. The outline of the perceptron structure is shown in Fig. 3. In this case, 225 descriptor values of space-neighborhood windows around a given window for two previous time steps are the inputs and the descriptor value of the central window considered for *t* time is the output. The connection pattern in the hidden layers can be seen in Fig. 4.

The same procedure described in section 2 for the fitting and prediction stages is followed here, but now, with non-linear equations.

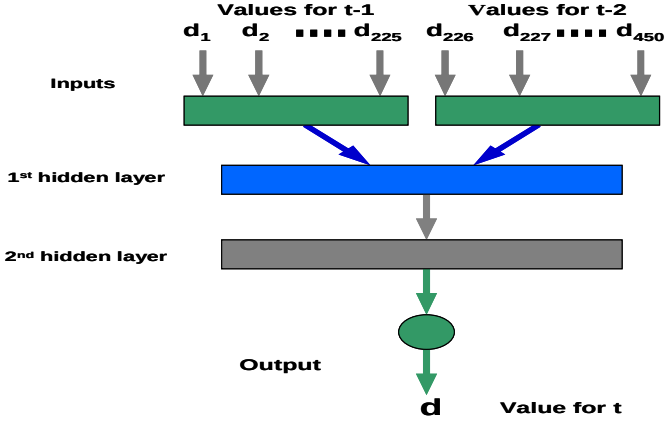


Fig. 3. Two hidden layer perceptron structure

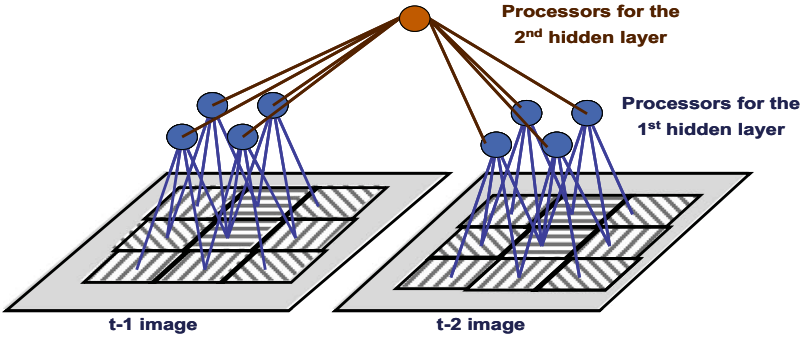


Fig. 4. Connection pattern in the hidden layers

4 Results

In this section, results obtained for both linear STAR-based- and neural network-model are described.

One image per day is available in our sample. Cross-validation techniques are used to decide the neighborhood size and topology of the models, obtaining square neighborhoods of 15x15 windows and two previous time steps. Both models are applied to predict images one time step ahead the last known image, that is, 24 h later.

Regarding the linear STAR-based model, considering the 225 space-neighborhood around a given window and substituting T=2 in Eq. (1), 450 parameters must be adjusted for every window of an image, except for a band around the boundaries of the same width as the neighborhood width, that is, seven windows. An iterative process is used to minimize the error involving a set of 150 images corresponding to different periods of the year.

Applying Eq. (2) to the 150 cases, one obtains the $(A)^{i,min}$ which minimizes ASQE. Then, substituting $(A)^{i,min}$ in Eq. (3), the forecasting method can be applied. Therefore, the descriptors of an image can be forecasted by knowing the same descriptors in a 24h- and 48h- image before.

The main difficulty found here was the redrawing of the predicted images in a classical isobar form. A visual characterization of the forecasted images is used in order to accomplish continuity. The real images to be compared with are also drawn in the same form.

For isobar redrawing out of window descriptors, a simple approach is followed, creating a set of parallel isobar lines rotated according to the orientation descriptor and spaced so as to achieve the isobar density descriptor. Continuous line drawing is not pursued.

Images of Figs. 5-8 and 9-12 show the obtained results for the isobar density and isobar orientation descriptors in a February and a June example respectively, for the both models studied. Prediction area is framed.

Reasonable results are obtained with the linear model. Considering the average isobar distance inside a window, average global errors are of about 25%. Regarding the isobar orientation inside a window, average global errors are less than 20%.

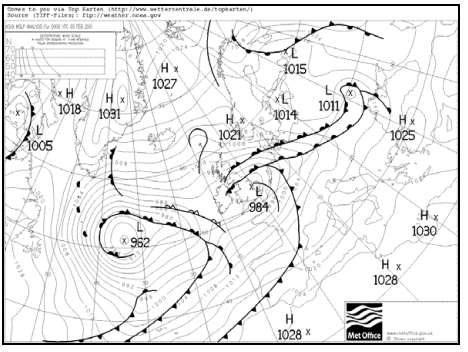


Fig. 5. Real image to be forecasted corresponding to 2001 5th February

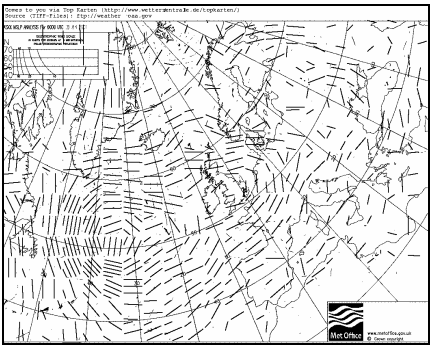


Fig. 6. Recreated image of Fig. 5

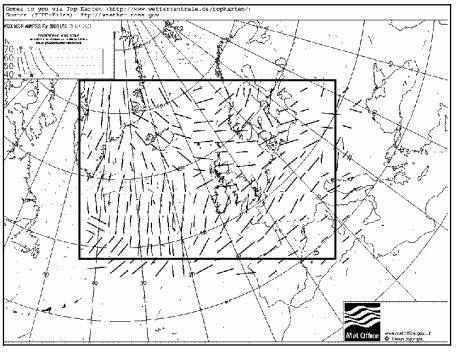


Fig. 7. Forecasted image obtained within our linear STAR-based model

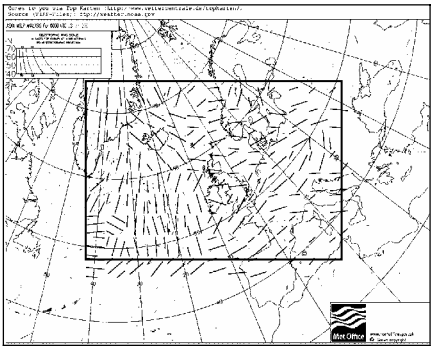


Fig. 8. Forecasted image obtained within our neural network mode

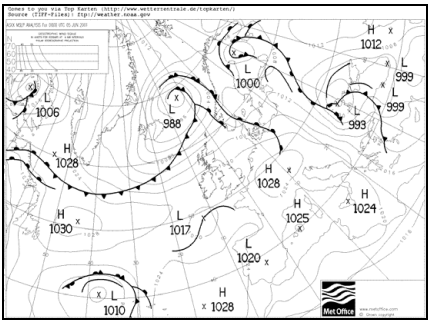


Fig. 9. Real image to be forecasted corresponding to 2001 5th June

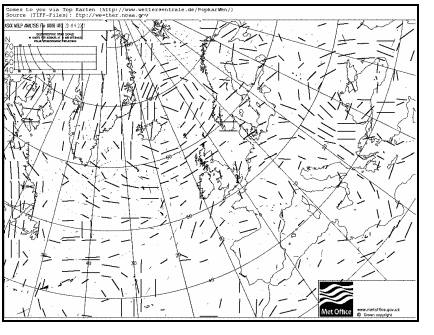


Fig. 10. Recreated image of Fig. 9

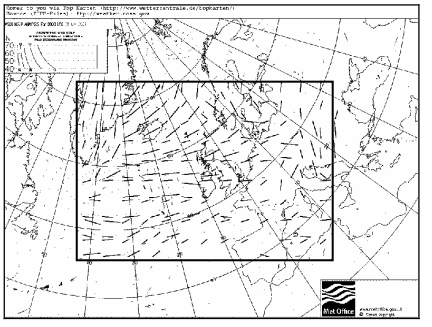


Fig. 11. Forecasted image obtained within our linear STAR-based model

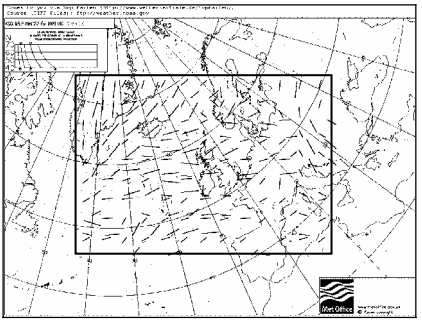


Fig. 12. Forecasted image obtained within our neural network model

Abnormal distributions for the residues are produced in the linear model and, as a consequence, a non-linear model is suggested. However, no substantial improvement is obtained with our neural network model and hence, this non-linear treatment is not justified versus the simpler linear model.

Regarding prediction for the centers of high and low pressures, bad results are obtained when working with H and L letters. The cross symbols could not be used because they are easily confused with any line crossing.

Other type of meteorological pressure maps are used elsewhere [4] for similar prediction purposes.

5 Conclusions

Our objective has been to study several approaches to forecasting the temporal evolution of images where optical flow techniques are not applicable. A significant point in these images is the absence of objects.

In the meteorological synoptic maps that we have worked with, possible object candidates like high and low pressure areas, move, change their shape, and may appear and disappear from one map to the next one.

The original images are split into adequate small windows where numerical descriptors are defined. Then, a window by window prediction is carried out for each numerical descriptor. The independent variables of the prediction function are the corresponding descriptor values of the window to predict and its spatial neighbours in previous time steps.

We have applied linear and non-linear models for the prediction purposes and discussed the obtained results. The prediction linear model used is based on a spatio-temporal autoregressive (STAR) framework. Regarding non-linear models, artificial neural networks have been chosen with standard fully connected two-layer perceptron structures.

In both models, the neighbourhood structure in space and time is obtained from cross-validation techniques. Forecasted images must be compared with the corresponding real images drawn in the same form.

Reasonable results are obtained within the linear model for the isobar density and isobar orientation descriptors. Considering the average isobar distance and average isobar orientation in a window, average global errors are of about 25% and 20%, respectively.

No substantial forecast improvement is obtained with the neural network model. Therefore, this non-linear treatment is not justified versus the simpler linear model.

Acknowledgments

The authors are deeply grateful to the Spanish Interministerial Board for Science and Technology (Comisión Interministerial de Ciencia y Tecnología CICYT) which supports Project TIC2002-01306, as a part of which this paper has been produced. They would also like to thank the EUVE (European Virtual Engineering) Technology Centre for their collaboration.

References

1. Bors, A., Pitas, I.: Prediction and tracking of moving objects in image sequences. IEEE Trans. Image Processing 8, 1441–1445 (2000)
2. Crespo, J.L., Bernardos, P., Zorrilla, M.E., Mora, E.: Preprocessing Phase in the PIETSI Project (Prediction of Time Evolution Images Using Intelligent Systems). In: Moreno-Díaz Jr., R., Pichler, F. (eds.) EUROCAST 2003. LNCS, vol. 2809, pp. 651–660. Springer, Heidelberg (2003)
3. Crespo, J.L., Bernardos, P., Zorrilla, M.E., Mora, E.: Meteorological Image Descriptors. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 101–110. Springer, Heidelberg (2005)
4. Crespo, J.L., Zorrilla, M.E., Bernardos, P., Mora, E.: A new image prediction model based on spatio-temporal techniques. The Visual Computer (2007), doi:10.1007/s00371-007-0114-y
5. Elnagar, A., Gupta, K.: Motion Prediction of Moving Objects Based on Autoregressive Model. IEEE Trans. on Systems, MAN, and Cybernetics — Part A: Systems and Humans 28(6), 803–810 (1998)
6. Hsiao, Y.T., Chuang, C.L., Lu, Y.L., Jiang, J.A.: Robust multiple objects tracking using image segmentation and trajectory estimation scheme in video frames. Image Vis. Comput. 24, 1123–1136 (2006)
7. Kehtarnavaz, N., Griswold, N.: Establishing collision zones for obstacles moving with uncertainty. Computer Vision, Graphics and Image Processing 49(1), 95–103 (1990)
8. Pece, A.E.C., Worrall, A.D.: A comparison between feature-based and EM-based contour tracking. Image Vis. Comput. 24(11), 1218–1232 (2006)
9. Pfeifer, P.E., Deutsch, S.J.: A three-stage iterative procedure for space-time modeling. Thechnometrics 22(1), 35–47 (1980)

Ontology-Based Modelling of Session Management Policies for Groupware Applications

Mario Anzures-García[1], Luz A. Sánchez-Gálvez[2], Miguel J. Hornos[2], and Patricia Paderewski-Rodríguez[2]

[1] Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla, 14 sur y avenida San Claudio. Ciudad Universitaria, San Manuel, 72570 Puebla, Mexico
manzures@siu.buap.mx, anzures@correo.ugr.es

[2] Departamento de Lenguajes y Sistemas Informáticos, E.T.S.I. Informática y de Telecomunicación, Universidad de Granada, C/ Periodista Saucedo Aranda, s/n, 18071 Granada, Spain
luzsg@correo.ugr.es, {patricia, mhornos}@ugr.es

Abstract. Groupware applications support group work and groups may perform geographically distributed with different roles and using different organizational structures. It is important that these applications do not provide only the needed infrastructure to support group work, but also grant some ways to change the group behaviour during a session. Therefore, this paper presents a conceptual model of the session management policies by means of an ontology, i.e. we model the form in which group work is organized. This ontology describes the concepts, the relations among them and the axioms that allow to specify and control the role changes that a user can play in a session and the changes in the rights/obligations of each role. In this model we have separated the mechanisms that consent to establishing a session (i.e. mechanisms to create the shared workspace) from the session management policies. Moreover, we can create templates of policies that can be employed to facilitate their reuse in runtime. In this way, we can affirm that this model smooths the dynamic group work.

1 Introduction

The CSCW (Computer Supported Cooperative Work) is defined as "a computer-based system that supports groups of people engaged in a common task (or goal) and provides an interface to a shared environment" [6]. The CSCW applications are also generally known as groupware or collaborative applications. Nowadays, this type of applications as well as providing an appropriate infrastructure to back group work, must support the dynamic structure of the organizations in runtime. Consequently, the groupware applications must provide mechanisms to back changes in the way in which the group work is carried out, as well as the different work styles of the organization.

The groupware applications offer a shared space to support the group work. Most applications of this type have done an abstraction of the group work context. Various

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 57 – 64, 2007.
© Springer-Verlag Berlin Heidelberg 2007

terms have been used to denote this abstraction, such as: conversations [11], locals [8], places [7], spaces [2], conferences [14, 15, 19], and sessions [4, 9]. In general, all these terms denote a set of geographically distributed individuals who share a common interest, that is to say carrying out ordinary tasks. We use the *session* word to denote the context of collaborative group work.

Some collaborative applications (such as: Groupkit [15], ANTS [9]) provide a mechanism to support a session called *session manager*. On the one hand, this mechanism allows to establish the session (i.e., it permits to establish the connection, to create and manage meetings, and to enable a user to join and leave a session using a simple user interface). On the other hand, this mechanism allows to define the called *session management policy* that states how sessions are organized to accomplish the group work (this facilitates the creation of different organizational structures that meet the particular needs of a group).

In general, collaborative applications do not separate the mechanisms to establish the shared group work context out from the session management policy. However, we think that this separation is important, since it will ease the support for changes in the group work and the reusing of the session management policy.

We present an ontology-based conceptual model of the session management policy in order to support and develop flexible sessions satisfactorily, so that they can adapt to the changes in the group work and to the different working styles of various groups. This model is part of our web service-based architectural proposal [1] that provides a suitable infrastructure for the development of groupware applications.

The rest of the paper is organized as follows. In Section 2, a brief introduction of the ontologies is given. In Section 3, we will present the session management policies. Section 4 describes the ontology-based modelling of the Session Management Policies. The final section will portray our conclusions and future work.

2 Ontologies

There are several definitions of ontology that have different connotations depending on the specific domain. In this paper, we will refer to Gruber's well-know definition [10] where an ontology is an explicit specification of a conceptualization. For Gruber, *conceptualization* is an abstract, simplified view of the world that we wish to represent for some purpose, by the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold them. Furthermore, *explicit specification* means that concepts and relations need to be couched by means of explicit names and definitions.

The ontology is presented as an organization resource and knowledge representation through an abstract model. This model of representation provides a common vocabulary of a domain and defines the meaning of the terms and the relations amongst them. Gruber identifies five kinds of components that are used to represent the knowledge of some domain: classes, relations, functions, instances and axioms [10]:

- *Classes* represent the (abstract or concrete) concepts of the domain to formalize. In an ontology, classes are organized in taxonomies, through inheritance mechanisms.
- *Relations* represent a type of interaction amongst the concepts of the domain. They are formally defined as any subset of a product of *n* sets (R: C_1 x C_2 x … x C_{n-1} x C_n).
- *Functions* are a special case of relations.
- *Instances* are used to represent the objects, elements or individuals of an ontology.
- *Axioms* are employed to model sentences that are always true. They are usually used to represent knowledge and also to verify the consistency of the ontology itself.

For us, in the domain of CSCW applications, the ontology provides a well-defined common and shared vocabulary, which supplies a set of concepts, relations and axioms to describe this domain in a formal way.

In CSCW environments, ontologies have mainly been used to model task analysis or sessions. Different concepts and terms such as group, role, actor, task, etc. have been used for the design of task analysis and sessions. Many of these terms are considered in our conceptual model. Moreover, semiformal methods (e.g. UML class diagrams, use cases, activity graphs, transition graphs, etc.) and formal ones (such as algebraic expressions) have also been applied to model the sessions. However, we have not found any model in the literature for the session management policies.

3 Session Management Policy

Policies are rules that govern the behaviour of a system in terms of the conditions under which predefined operations or actions can be invoked [12, 13, 17]. Policies are persistent since a policy defines a future action to be carried out, or repeated actions, or actions related to the maintenance of a condition. It should be noted that during the runtime of a groupware application, it must be possible to establish new policies and to modify the existing ones.

Groupware applications are typically large-scale systems, which require solutions in accordance with the behaviour change of the system, i.e. self-adapting and dynamic solutions. A promising solution to manage distributed systems (and in particular groupware applications) is the policy-based management [18]. It allows to dynamically specify adaptable management strategies that can be easily modified in order to change the management approach without recoding or stopping the system.

The CSCW domain is mainly focused on coordination and security management policies. The coordination policies facilitate the interaction among users and among users and the shared resources, and also avoid conflicts leading to information inconsistencies. Examples of this type of policies are applied to: access control [16], concurrency control [5] and floor control [3]. The security policies are necessary to

outline the actions to be taken when security violations, such as a series of login failures occur to a particular user, or an attack to the system is detected.

With respect to the session management policies, in the literature we can find that session managers can implement and provide a broad variety of policies to users. We mention two of the most important ones:

- *A brainstorming policy*, which is designed for informal collaborations among equal hierarchy users, i.e. users with equal status. For this reason, users must avail themselves of voting tools to decide who establishes turns for their participation or if it is determined on a first-come-first-participating basis. This policy functions similarly to instant messaging applications.
- *A moderate policy*, which is designed for a structured group work controlled by a single person, the so-called moderator. The latter has a better status than other users, that is why s/he controls and coordinates the session and also establishes turns for user participation.

In our work, policies are used to dictate the organizational structure of the group work, in such a way that they allow the establishment of the session control (which determines who authorizes users registration, how interaction is carried out among users, how turns for users participation are defined) and the support of changes (which specify how they modify the user role, the role permissions and/or the current policy) in runtime. Each session presents a group organizational structure at a given moment and each session management policy is defined in the design stage according to the ontology we will present in the next section.

4 Ontology-Based Conceptual Model

In this section we present an ontology for conceptual modelling of the session management policies for groupware applications. The ontology has been used to model these policies since:

- It provides a taxonomic structure of the domain (group organizational structure), which facilitates the representation of the hierarchical character of the organization.
- It admits an explicit specification of the elements of the domain by means of concepts, relations and axioms. The first two components of our ontology facilitate the modelling of the session management policies. The axioms specify the conditions (or restrictions) to be taken into account to carry out the suitable changes in accordance with the group dynamics. Accordingly, this ontology considers the dynamical aspects (i.e., changes in the group size or in the group work style) of the group organizational structure in runtime.

4.1 Ontology Concepts for the Session Management Policies

We consider that the group organizational structure is governed by a specific policy, which determines how the group work is organized by means of the roles that users

can play. These roles establish the set of rights/obligations and the position (status) into the group organizational structure, as well as the tasks that can be carried out. Users can play one o more roles and employ the shared resources to carry out tasks.

In our ontology, we propose the following concepts or classes (see Figure 1):

- ***Group organizational structure***, which is governed by a policy that defines how the group work is organized by a hierarchical structure, as it happens in universities, offices, departments, and hospitals. Sometimes it is necessary to delineate the group size, which indicates the maximum number of users in the organization.
- ***Policy***, which establishes a configuration of the group organizational structure in accordance with the status assigned and the rights/obligations established for each role defined.
- ***Role***, which contains a set of rights and obligations related to a position (status) in an organisation, such as university chancellor, department director, network operator, and/or teacher. Each role defined by the policy must be played at least by a user so that the policy can operate in a suitable way. For example, in the moderate policy at least one user must play the *moderator* role and no less than another one must play the *participant* role.
- ***Status***, which defines the role authority according to the position of this inside an organisation. Since the status gives a set of privileges with respect of other users, it allows to establish a hierarchy structure. The status determines who is the user that controls or manages the organization in a given moment and who must take his/her position in case that a user with the highest status is not present.
- ***Right/Obligation***, which describes the rights and/or obligations in accordance with the role. The rights/obligations are defined in terms of the position role rather than in terms of the person that is playing the corresponding, so that the applied policy does not have to be changed when people are reassigned to different positions within the organisation. These rights/obligations constrain the user actions in the organization; he/she must interact with other users and carry out collaborative tasks. For example, the professor has to teach and research.
- ***User***, who is a person that takes part in the group work. Users participate playing roles that allow them to carry out collaborative tasks in accordance with their rights/obligations.
- ***Task***, which is an activity carried out by one or more users to achieve a given goal in a certain time period. A task can be done by one or more users and sometimes can be divided into other simpler tasks (subtasks).
- ***Shared Resource***, which represents the resources used by the users to carry out the tasks allowing to achieve the common aims defined inside the group organizational structure.

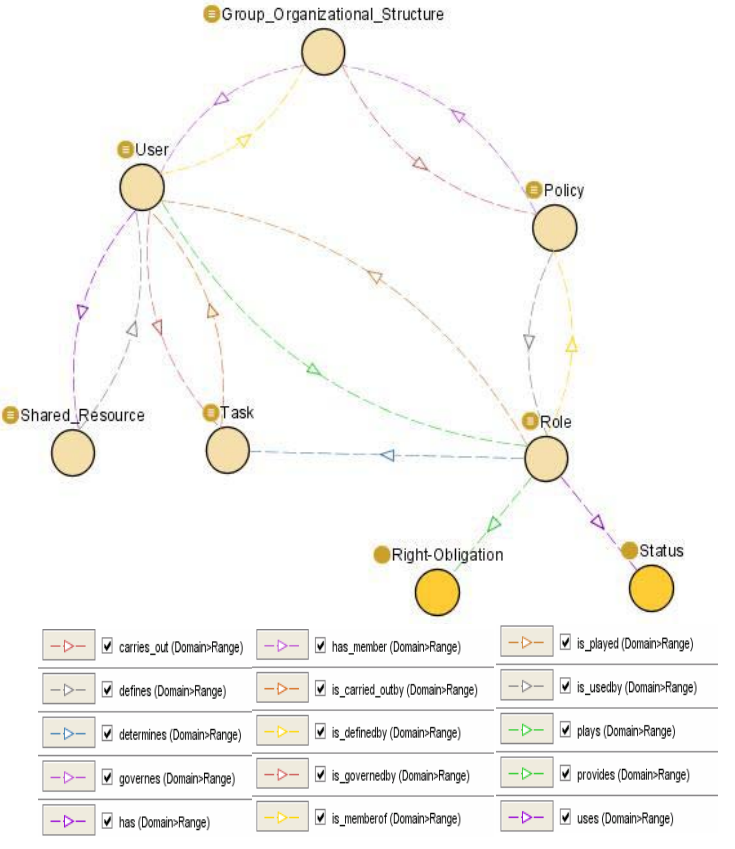


Fig. 1. Conceptual Model of the Session Management Policy based on an OWL[1] Ontology. Figure generated by Jambalaya[2] plug-in for Protégé[3].

4.2 Ontology Relations for the Session Management Policies

The concepts defined in the previous subsection are associated by the following relations, as Figure 1 shows:

- *is_governed_by(Group_Organizational_Structure, Policy)*, which specifies that a group organizational structure is governed by a policy at a given moment.
- *is_member_of(User, Group_Organizational_Structure)*, which defines that the user is member of the group organizational structure.
- *defines(Policy, Role)*, which specifies that the policy defines the roles that users can play.

[1] Online at: http://www.w3.org/2004/OWL/
[2] Online at: http://www.thechiselgroup.org/jambalaya
[3] Online at: http://protege.stanford.edu/

- *plays(User, Role)*, which indicates that the user can play roles of an organizational structure.
- *has(Role, Status)*, which determines that the role has a status inside the group organizational structure.
- *provides(Role, Right-Obligation)*, which specifies that the role provides a set of rights/obligations.
- *determines(Role, Task)*, which indicates that each role determines the tasks that a user playing it can carry out inside the group organizational structure.
- *uses(User, Shared_Resource)*, which defines that the user employs the shared resources to achieve a common goal.
- *carries_out(User, Task)*, which specifies that the user carries out one o more collaborative tasks of the organizational structure.

4.3 Ontology Restrictions for the Session Management Policies

In this section, we will present some restrictions of our ontology (see Figure 1):

- A group organizational structure is only governed by a policy.
- A group organizational structure has at least two users.
- Each policy defines at least one role.
- Each role has to be played by at least one user.
- Each task has to be carried out by at least one user.

5 Conclusions and Future Work

The main purpose of this paper has been the presentation of a novel ontology-based model to support the specification of session management policies for groupware applications. This model takes into account the dynamic aspects of the group organizational structure allowing to adapt the session to the following changes in the:

- *group size*, since it allows a user to join or leave a session, i.e. the policy assigns a role to each user who joins a session or assigns the role played by a user that leaves the session to another user when that role is not played by any user.
- *user role*, since it facilitates to change the user role in runtime. The new role is assigned according to the roles that have been defined by the policy.
- *rights/obligations*, since it enables to modify them in accordance with the role position inside the organization rather than with the user playing the corresponding role.
- *policy*, since it is possible to define different policies in accordance with the model defined here. In this way, a new policy can be chosen in runtime.

Moreover, the ontology allows us to define the necessary requirements to model the session management policies and to reuse these policies in several groups with different needs and work styles.

Our future work will be focused on the development of a catalogue of policies that can be chosen by the user either at the beginning of the session or in runtime.

References

1. Anzures-García, M., Hornos, M.J., Paderewski, P.: Development of extensible and flexible collaborative applications using a web service-based architecture. LNCS, vol. 4401, pp. 66–80. Springer, Heidelberg (2007)
2. Beaudouin-Lafon, M.: Beyond the workstation: Mediaspaces and augmented reality. In: Cockton, G., Draper, S.W., Weir, G.R.S. (eds.) Proceedings of HCI. People and computers, vol. 9, pp. 9–18. Cambridge University Press, Cambridge (1994)
3. Dommel, H.P., Garcia-Luna-Aceves, J.J.: Floor Control for Multimedia Conferencing and Collaboration. ACM Multimedia 5(1), 23–38 (1997)
4. Edwards, W.K.: Session management for collaborative applications. In: Proceedings CSCW, pp. 323–330 (1994)
5. Ellis, C.A., Gibbs, S.J.: Concurrency control in groupware systems. In: Proceedings ACM SIGMOD, ACM Press, New York (1989)
6. Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware: Some issues and experiences. Communications of the ACM 34(1), 39–58 (1991)
7. Fitzpatrick, G., Kaplan, S.M., Mansfield, T.: Physical spaces, virtual places and social worlds: A study of work in the virtual. In: Proceedings CSCW, pp. 334–343 (1996)
8. Fitzpatrick, G., Kaplan, S.M., Tolone, W.J.: Work, locales and distributed social worlds. In: proceedings ECSCW, pp. 1–16 (1995)
9. García, P., Gómez, A.: ANTS Framework for Cooperative Work Environments. Computer 36(3), 56–62 (2003)
10. Gruber, T.R.: Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human Computer Studies 43(5/6), 907–928 (1995)
11. Kaplan, S.M., Caroll, A.M.: Supporting collaborative processes with conversation builder. Computer Communications 15(8), 489–501 (1992)
12. Moffett, J.D., Sloman, M.S.: The representation of policies as system objects. In: Proceedings SIGOIS, vol. 12(1), pp. 71–84 (1991)
13. Moffett, J.D., Sloman, M.S.: Policy hierarchies for distributed systems management. IEEE Journal on Selected Areas in Communications 11(9), 1404–1414 (1993)
14. Rajan, S., Venkat, R.P., Vin, H.M.: A formal basis for structured multimedia collaborations. In: Proceedings of the 2nd IEEE International Conference on Multimedia Computing and Systems, pp. 194–201. IEEE Computer Society Press, Los Alamitos (1995)
15. Roseman, M., Greenberg, S.: Building real time groupware with GroupKit, a groupware toolkit. ACM Transactions on Computer-Human Interaction 3(1), 66–106 (1996)
16. Shen, H., Dewan, P.: Access control for collaborative environments. In: Proceedings CSCW of ACM, pp. 51–58. ACM Press, New York (1992)
17. Sloman, M.: Policy driven management for distributed systems. J. Network System Management 2(3), 33–60 (1994)
18. Sloman, M.: IDSM & SysMan common architecture, Domain & Policy Concepts. In: Proceedings SMDS (1995)
19. Venkat, R.P., Vin, H.M.: Multimedia conferencing as a universal paradigm for collaboration. In: Kjelldahl, L. (ed.) Multimedia: Systems, interaction and application, 1st Eurographics Workshop, pp. 173–185. Springer, Heidelberg (1991)

Computer Aided Development and Simulation of a Remote Control VTOL Tiltwing Model Aircraft

A.P. de Madrid,[1] P. Ruipérez, M. Romero, C. Mañoso, and R. Hernández

Escuela Técnica Superior de Ingeniería Informática. UNED.
Departamento de Sistemas de Comunicación y Control
Juan del Rosal, 16. 28040 – Madrid, Spain
angel@scc.uned.es

Abstract. This paper describes the computer aided development of a remote control tiltwing scale aircraft. Tiltwings —even scale models— are difficult to fly by human pilots due to their instability and the different flying modes they exhibit. For this reason a complex flight simulation and training environment and an onboard control system (fly-by-wire) that increases stability have been developed. Modular design, both for hardware and software, has been kept in mind in order to modify the prototypes easily and use them as test-beds for new concepts and configurations.

Keywords: VTOL, Tiltwings, Tiltrotors, Fly-by-wire, Control systems, Remote control, Flight simulation, Computer tools.

1 Introduction

VTOL stands for Vertical Take-Off and Landing. Since the 1930s, many efforts have been done to develop aircrafts that exhibit this capacity, but very few prototypes have led to operational aircrafts (helicopters, autogyros or balloons are not considered VTOL) [1].

Tiltrotors are one of the most important types of practical VTOL aircrafts. They combine the vertical lift capability of a helicopter with the speed of a turboprop airplane. They use tiltable propellers, or proprotors, for lift and propulsion. For vertical flight the proprotors are angled to direct their thrust downwards, providing lift. As the craft gains speed, the proprotors are slowly tilted forward, eventually becoming perpendicular to the ground. In this mode the wing provides the lift (Fig. 1). The tiltrotor's advantage is significantly greater speed than a helicopter [2].

Tiltwings are similar to tiltrotors but, whereas a tiltrotor rotates the prop from axial to dorsal, a tiltwing rotates the entire wing, not just the nacelles [3].

Recent operational tiltrotors are the military V-22 Osprey (first flight in 1989) [4] and the civil Bell/Agusta BA609 [5]. The 6th European Framework Program will

[1] Corresponding author.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 65 – 72, 2007.
© Springer-Verlag Berlin Heidelberg 2007

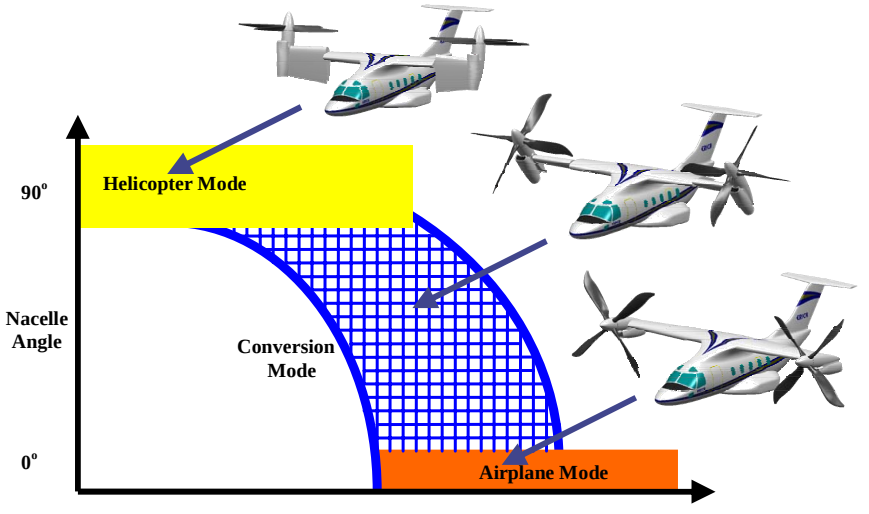


Fig. 1. Tiltwing and tiltrotors: The different flying modes. The figure depicts Erica, the future European transport.

contribute to develop the advanced tiltrotor Erica within a large partnership [6], [7]. (In fact, Erica is neither a tiltwing nor a tiltrotor: It is a "hybrid" model, as the central wing segment does not rotate, as shown in Fig. 1.)

Since 2005 the authors have been collaborating with aeroVertical, a group focused on hi-tech remote control (RC) planes research and development [8]. As a result, several RC tiltwing prototypes have been built and tested and a simulation and training environment has been developed.

These prototypes have to be piloted as any other RC model, so an average human pilot should be able to flight them using a conventional RC emitter (we are not focusing on designing an unmanned aerial vehicle —UAV— yet). However, tiltrotors and tiltwings are highly unstable, especially during the hovering and conversion flight modes. Thus an onboard automatic flight control system that increases stability, i.e., a fly-by-wire control system [9], is needed.

Pilot training has also been kept in mind. The pilot has to become familiar with the different flying modes, maneuvers, conversions, etc., before flying the actual prototype. For this reason, a photorealistic and physically accurate flight simulator has also been developed.

Due to its complex nature, all the design process has involved an intensive use of CAD and 3D software.

This paper is organized as follows: Section 2 describes the RC prototypes. Avionics is described in section 3. Two virtual models are presented in section 4. Section 5 is dedicated to the simulation and training environment. Finally, section 6 draws the main conclusions of this paper.

2 The Prototypes

From the point of view of their components and manufacture, the prototypes are conventional RC models with an onboard computer. In order to keep the cost as low as possible, the prototypes are built using off-the-shelf components. This means using low-cost motors, propellers, sensors (gyroscopes), receptors and components that are widely used for building RC models, as well as low cost microprocessors needed by the fly-by-wire control system.

These prototypes are intended to be test beds for testing different airframes and wing configurations, as well as different control strategies. For this reason, modularity and versatility have been kept in mind during the whole design process. In this way, it is easy to change a wing, not because it is broken (on the other hand, a feasible possibility), but because a different configuration is going to be tested, or an advanced robust control strategy can be downloaded instead of a more conventional controller.

Due to this modularity, we cannot talk about “the prototype”. Several different configurations have been actually built and tested, including true tiltwings and “hybrid” airplanes similar to Erica.

A 3 degrees-of-freedom test-bench (Fig. 2) is also available. It is used to test the prototype behavior before free-flight. It is a very useful tool when a new concept is tested for the first time.



Fig. 2. The 3 degrees-of-freedom test-bench. In this tethered configuration, the prototype is tested with safety before free-flight. The figure shows a tiltwing prototype.

3 Avionics

Avionics is the set of electronic devices used on the aircraft. All RC models share some common components such as a receptor and some servos. The described VTOL prototypes need some sensors, a fly-by-wire control system and a data-logger as well (Fig. 3).

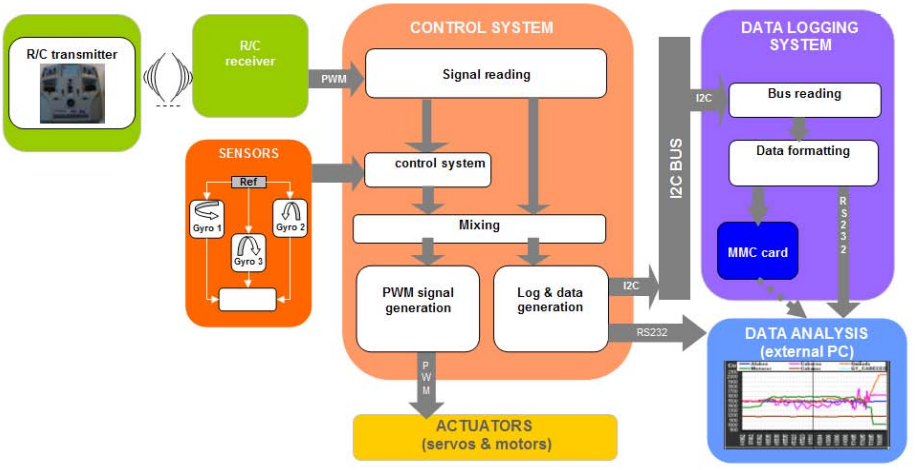


Fig. 3. Functional schema of electronics, control and data logging. The human pilot's commands are sent from the block labeled "R/C Emitter".



Fig. 4. Hovering

3.1 Sensors

Early models just carry three gyroscopes. Combined with a quite simple control system (see Section 3.2), they provide stabilization enough to fly the prototype in hovering (Fig. 4).

However, conversion modes require more information that is not provided by the gyroscopes. Nowadays, inertial measurement units —IMUs—, that combine gyroscopes, accelerometers and magnetometers to track how the airplane is moving, are being tested. This information can be integrated with altimeters, GPS and optical sensors for giving a more accurate estimation of the prototype attitude.

3.2 Fly-by-Wire Control System

As it was mentioned above, tiltwings/tiltrotors are highly unstable systems. In order to achieve a system easier to fly by an average human pilot, a fly-by-wire control system is needed. It reads the pilot's commands (through the RC receptor), the attitude (from the sensors) and sends control signals to the actuators (servos and motors). In this way, the control system interprets the pilot's commands and decides how to fulfill his/her orders.

The control system is nowadays implemented in simple microprocessors of the PIC18F family [10]. However, more computational power is needed to control all the different flight and conversion modes. Due to the modular design, different processors can be boarded with ease.

3.3 Data Logging

Data analysis is needed to complete the design cycle. During an experience (a free-flight or a test-bench test), all relevant data are stored in a conventional flash memory card for offline analysis. In the test-bench, data can be read from a computer through two RS232 serial links in real time. Wireless links exist that could be incorporated to the prototype in the future for free flight online monitoring and analysis.

4 The Virtual Models

One of the objectives of the design process is obtaining an accurate virtual model of the prototype. In fact, two models have been developed: A CAD/visualization model and a simulation model. Each one depicts a different aspect of reality:

- The CAD/visualization model includes every part, even the movable ones (tiltable wings, propellers, control surfaces, etc.), of the prototype with a high level of realism (Fig. 5). As the weight is a limiting factor, this model is used to study the effect of different materials, electrical/mechanical components, etc. on weight. It also provides some physical information needed by the simulation model: total weight, center of gravity position and its change during the different flying modes, components of the inertia tensor, etc. It is exported to a flight simulation environment (FlightGear, an open-source simulator [11]) for visualization purposes.
- The simulation model has been built with Matlab/Simulink [12], an industry and academia design, analysis and simulation standard tool. Simulations are needed for evaluating the flying qualities of the prototype, designing control systems, analyzing and improving the conversion modes, etc. Matlab/Simulink is also used as the simulation engine of the flight simulation environment.



Fig. 5. From concept to reality. This figure depicts the virtual model and the actual airplane of an early tiltwing prototype with ducted propellers. Due to vibration problems, current prototypes use propellers that are not ducted.

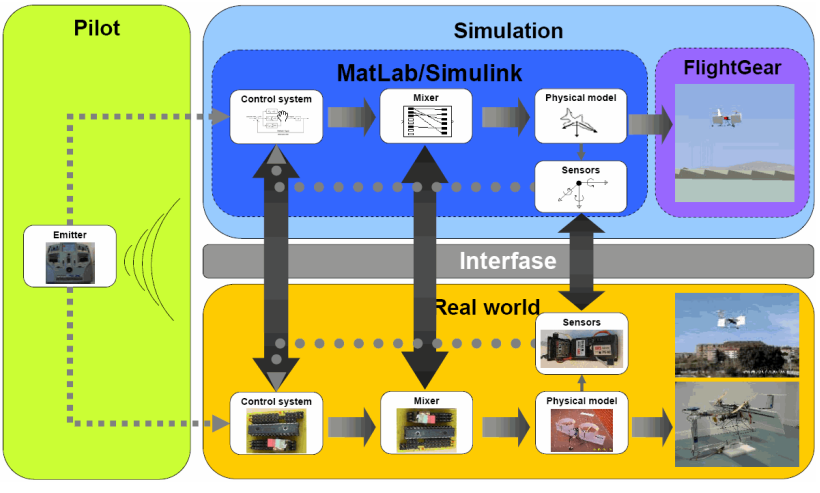


Fig. 6. Simulation and training environment

5 Simulation and Training Environment

The simulation and training environment was designed using a well defined set of components and a flexible architecture, making optimum use of adaptability and modularity concepts.

The visualization and simulation models meet here with a high degree of realism:

- The visual realism of FlightGear, and
- The accurate simulations of Matlab/Simulink.

In this way, the real world and the simulation tools are merged into a single environment (Fig. 6) that makes possible:

- Predict or simulate the model behavior when its parameters change (control surfaces, CG position, wing profiles, motor/wings optimal rotation axis, etc.) or radical modifications in the design concepts (tiltrotor, tiltwing, hybrid configurations, etc.) are introduced.

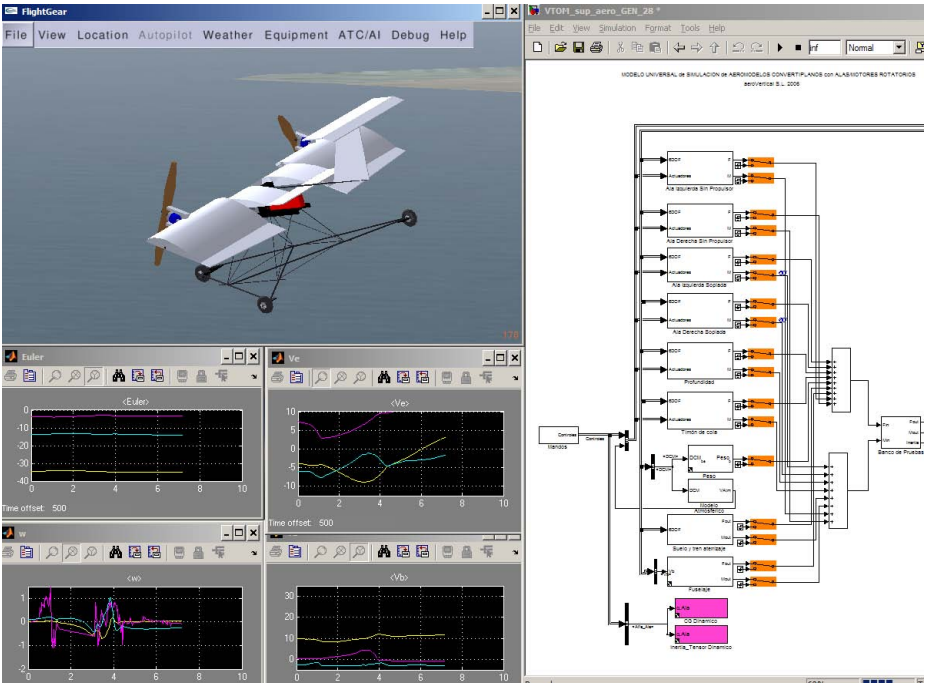


Fig. 7. Typical windows arrangement with FlightGear (*top left*) and Matlab/Simulink (*bottom left and right*) windows opened. All windows are refreshed in real time.

- Analyze and test the flight envelopes for the different flight modes. Predict where in the flight envelope conversions are possible and find out how to perform them.
- Test and validate hardware and software sensors, onboard software, control algorithms, etc. with no risk for the actual prototype.
- Train the pilot to manage the flight modes and critical maneuvers, with no risk of crashing the prototype during the learning process. Skillful pilots can even improve risky or difficult maneuvers and write the manual for novel pilots.
- Just have fun flying around.

Fig. 7 shows a typical windows arrangement. It can be modified (just moving the windows around, resizing them or opening new ones) for performing different experiences and visualizing all relevant data in real time.

6 Conclusions

This paper has described the computer aided development of a remote control tiltwing scale aircraft. The project involves not only the aircraft prototypes but a complete simulation and training environment. As a whole, it is an excellent tool for teaching or doing research on aeronautics, control systems, modeling and simulation, etc., as well as a challenging and enjoyable airplane to fly (both in simulation or the actual prototype).

Our current research includes the development of new control strategies and the integration of more accurate and powerful systems, including new sensors (altimeter, GPS, video cameras…), IMUs, servos and microprocessors. By doing so, the prototype could be used as an UAV in the future.

Acknowledgments. The authors wish to thank aeroVertical for their collaboration in this project and building the prototypes. This work was supported by the Spanish *Ministerio de Educación y Ciencia*, under project ref. DPI2004-05903.

References

1. Hirschberg, M.J.: An Overview of the History of Vertical and/or Short Take-Off and Landing (V/STOL) Aircraft, http://www.vstol.org/
2. “Tiltrotor”: Wikipedia, the free encyclopedia, http://en.wikipedia.org/
3. “Tiltwing”: Wikipedia, the free encyclopedia, http://en.wikipedia.org/
4. Bell Boeing V-22 Osprey, http://www.boeing.com/rotorcraft/military/v22/
5. Bell/Agusta BA609, http://www.bellagusta.com/
6. Cicale, M.: Erica: The European Tiltrotor, Europe/US International Aviation Safety Conference, Cologne (June 7-9, 2005), http://www.easa.eu.int/conference2005/ presentations/ day2/New_Technologies/erica_easa.pdf.
7. Pagnano, G.: Il convertiplano ERICA: evoluzione della ricerca nei Programmi Quadro europei. CRUI Viaggio della Ricerca in Italia Milano 17 giugno (2005), http://www.crui.it/ crui/milano_7pq/CRUI170605_pagnano.ppt
8. http://www.aerovertical.com/
9. “Fly-by-wire”. Wikipedia, the free encyclopedia, http://en.wikipedia.org/wiki/Fly_by_wire#Fly-by-wire
10. PIC18, MCU, http://www.microchip.com/
11. FlightGear, http://www.flightgear.org/
12. The Mathworks, http://www.mathworks.com/

A Simulation Study of New Security Schemes in Mobile Ad-Hoc NETworks*

P. Caballero-Gil[1], C. Caballero-Gil, J. Molina-Gil, and A. Quesada-Arencibia[2]

[1] Department of Statistics, Operations Research and Computing. University of La Laguna. 38271 La Laguna. Tenerife. Spain
pcaballe@ull.es
[2] Department of Informatics and Systems. University of Las Palmas de Gran Canaria. 35017 Las Palmas de Gran Canaria. Gran Canaria. Spain
aquesada@dis.ulpgc.es

Abstract. This work provides both a new global authentication system for Mobile Ad-hoc NETworks and a study of its simulation with NS-2. The proposed scheme is constructed in a self-organizing manner, which implies the fulfillment of most requirements for this type of networks, such as adaptation to the changing topology of the network, public availability of broadcast communications and strength of access control.

Keywords. Authentication, Access Control, MANETs, Simulation.

1 Introduction

Authentication, which guarantees the right identities of users, is the most remarkable security aspect in any system because all the remaining characteristics depend totally on it. Weak authentication schemes based on maximum-disclosure proofs with secret time-invariant passwords have a major security concern, which is the possible eavesdropping and subsequent replay of secret passwords. Two well-known solutions to this problem are variable passwords and minimum-disclosure proofs. The protocol here proposed combines both concepts in order to define a strong authentication scheme for Mobile Ad-hoc Networks.

Mobile Ad-hoc NETworks (MANETs) are autonomous wireless networks formed by mobile nodes. Applications for MANETs are diverse, ranging from small and static networks constrained by power sources, to large-scale and highly dynamic networks. Wired networks mainly use a Certificate Authority (CA) to solve the authentication problem. However, authentication in MANETs is more difficult due to reasons such as limited physical protection of broadcast medium, frequent route changes caused by mobility, lack of a structured hierarchy, etc [6].

Many authentication protocols have been recently proposed for ad-hoc networks [1] [4] [5]. However, the design of a scheme that fulfills the requirements for this type of networks continues being considered an open question. One of the

* Research partially supported by the Spanish Ministry of Education and Science and the European FEDER Fund under Project SEG2004-04352-C04-03.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 73–81, 2007.
© Springer-Verlag Berlin Heidelberg 2007

approaches found in the bibliography uses a Trusted Third Party (TTP) to guarantee the validity of all nodes identities. Another identification paradigms that have been used in wireless ad-hoc networks are the notions of chain of trust and location-limited authentication. The special nature of ad-hoc networks suggests that such traditional approaches may not be appropriate. This work proposes a different type of scheme based on the cryptographic primitive of Zero-Knowledge Proofs (ZKPs). Until now few publications have mentioned the proposal of authentication systems for ad-hoc networks using ZKPs [2] [7], and none of them has dealt with the related problem of topology changes in the network.

This work is organized as follows. The following section provides a complete description of the new proposal, including notation and specific details about network initialization, node insertion, access control, proofs of life and node deletion. The analysis of a full simulation through NS-2 is given in Section 3. Finally, some conclusions and open questions complete the paper.

2 Proposal

The proposal was designed as an authentication scheme for group membership. Consequently, robust methods to insert and to delete nodes, as well as to allow the access only for legitimate members of the group are provided. In particular, the access control scheme is based on the general scheme of Zero-Knowledge Proof introduced in [3], for the particular case of the Hamiltonian Cycle Problem (HCP). A hamiltonian cycle of a graph is a cycle that visits each vertex exactly once and returns to the starting vertex. Determining whether such cycles exist in a graph is an NP-complete problem. Such a problem was chosen for the design because the upgrade of a solution due to an insertion or a deletion of a vertex in the graph does not demand a great computational effort. Such operations will be frequent in our implementation due to the high dynamism of MANETs.

One of the key points of the proposed scheme is the use of a chat application through broadcast that makes it possible for on-line legitimate nodes to send a message to all on-line legitimate nodes and receive their answers. Although secrecy is not necessary for those chat messages because they are useless for non-legitimate nodes, since they contain data necessary for updating authentication information, it is required that only on-line legitimate nodes may execute the chat application. The whole information received through the chat application during an interval of time must be stored by each on-line node in a FIFO queue. Such information will allow the updating of the authentication information both for it and for all off-line legitimate nodes whose access to the on-line state is authorized by that on-line node. The length of such a period is an essential parameter because it states both the maximum off-line time allowed for any legitimate node, and the frequency of broadcasts of proofs of life. Consequently, such a parameter should be previously agreed among all the legitimate nodes. A remarkable aspect of the proposal is that no adversary is able to steal any meaningful information even if it reads all the information sent through the chat application, or if it eavesdrops all the data exchanged during an access control.

The following sub-sections give, respectively, a description of the used notation and specific details about network initialization, node insertion, access control, proofs of life and node deletion.

2.1 Notation

- $G_t = (V_t, E_t)$ denotes the undirected graph used at stage t.
- $v_i \in V_t$ represents both a vertex of the graph and a legitimate node.
- $n = |V_t|$ is the order of G_t, equal to the number of legitimate nodes.
- $N_{G_t}(v_i)$ denotes the set of neighbours of node v_i in the graph G_t.
- $\Pi(V_t)$ represents a random permutation over the vertex set V_t
- $\Pi(G_t)$ denotes the graph isomorphic to G_t corresponding to $\Pi(V_t)$.
- $c \in_r C$ indicates that an element c is chosen at random from a set C.
- HC_t designates the hamiltonian cycle used at stage t.
- $\Pi(HC_t)$ represents the hamiltonian cycle HC_t in the graph $\Pi(G_t)$.
- $N_{HC_t}(v_i)$ denotes the neighbours of node v_i in the hamiltonian cycle HC_t.
- S and A stand for the supplicant and the authenticator, respectively, both during an insertion phase and during an access control.
- $S \rightleftharpoons A$ symbolizes when node S contacts A.
- $A \leftrightarrow S : information$ means that A and S agree on $information$
- $A \overset{s}{\rightarrow} S : information$ means that A sends $information$ to S through a secure channel.
- $A \overset{o}{\rightarrow} S : information$ means that A sends $information$ to S through an open channel.
- $A \overset{b}{\rightarrow} network : information$ represents when A broadcasts $information$ to all on-line legitimate nodes.
- $A \overset{b}{\leftrightarrow} network : information$ denotes a two-step procedure where A broadcasts $information$ to all on-line legitimate nodes and receives their answers.
- h stands for a public hash function.
- T denotes the threshold length of the off-line period for legitimate nodes.

2.2 Network Initialization

This phase consists in the joint definition of the graph used within the protocol. After this phase, each legitimate node will know an initial and jointly generated solution of the HCP in such a graph. The set of vertices of the graph corresponds exactly to the set of nodes of the network during the whole life-cycle of the network. The first step consists of the joint and secret generation of a random permutation Π of such a set. Then, the partial graph formed by the hamiltonian cycle HC_0 corresponding to such a permutation, is completed by adding n groups of $\frac{2m}{n}$ edges, in order to produce the initial edge set E_0.

Initialization Algorithm

Input: V_0, with $|V_0| = n$

1. The n nodes of the network generate jointly, secretly and randomly the cycle $HC_0 = \Pi(V_0)$.

2. Each node $v_i \in V_0$ builds the set $N_{G_0}(i) = \{\{v_j \in_r V_0\} \cup N_{HC_0}(i)\}$ with $|N_{G_0}(i)| = \frac{2m}{n}$.
3. Each node broadcasts $v_i \xrightarrow{b} network : N_{G_0}(i)$
4. Each node merges $E_0 = \bigcup_{i=1,2,...,n} \{(v_i, v_j) : v_j \in N_{G_0}(i)\}$

Output: $G_0 = (V_0, E_0)$, with $|E_0| = m$

2.3 Node Insertion

When a non-legitimate node S wants to become a member of the network, it has to convince a legitimate node A to proceed with its insertion. Then, A should assign to S a vertex number v_i not yet assigned, broadcast this number to all on-line legitimate nodes in order to prevent another identical insertion, and receive their answers. If A receives more than $n/2$ answers, A chooses an upgrade of the hamiltonian cycle HC_t by selecting at random two neighbour vertexes v_j and v_k in order to insert v_i between them, chooses at random a set of $\frac{2m}{n} - 2$ nodes in V_t such that none of them are neighbours in HC_t, and broadcasts the set of neighbours $N_{G_{t+1}}(v_i)$ of S in the new graph G_{t+1} to all on-line legitimate nodes. Each time a node receives an updating of the graph, it has to identify the unique position in the cycle where the new node can be inserted according to the new edge set E_{t+1}. At the same time, A must send to S both G_{t+1} and HC_{t+1}.

Insertion Algorithm

Input: At stage t a supplicant node S wants to become a member of the network.

1. $S \rightleftharpoons A$ and node S convinces node A to accept its entrance to the network.
2. A assigns to S the vertex number v_i such that $i = min\{l : v_l \notin V_t\}$
3. A broadcasts $A \xleftrightarrow{b} network : v_i$
4. – If A receives less than $n/2$ answers, she stops the insertion procedure.
   – Otherwise:
   (a) A chooses at random $\{v_j \in_r V_t, v_k \in_r N_{CH_t}(v_j)\}$
   (b) A chooses at random $N_{G_{t+1}}(v_i) = \{v_j, v_k\} \cup \{w_1, w_2, ..., w_{\frac{2m}{n}-2} \in_r V_t$ such that $\forall w_{l_1}, w_{l_2} : w_{l_1} \notin N_{CH_t}(w_{l_2})\}$
   (c) A broadcasts $A \xrightarrow{b} network : N_{G_{t+1}}(v_i)$
   (d) Each on-line node computes $V_{t+1} = V_t \cup \{v_i\}$, $E_{t+1} = E_t \cup N_{G_{t+1}}(v_i)$ and $HC_{t+1} = \{HC_t \setminus (v_j, v_k)\} \cup \{(v_j, v_i) \cup (v_i, v_k)\}$
   (e) A sends openly $A \xrightarrow{o} v_i : G_{t+1}$
   (f) A sends securely $A \xrightarrow{s} v_i : HC_{t+1}$

Output: The supplicant node S is a legitimate member of the network.

2.4 Access Control

If a legitimate node S has been off-line or out-of-coverage from stage t and wants to connect on-line at stage r, it has to contact a legitimate on-line node A who checks whether the off-line period of S is not greater than T. Then, S

has to be authenticated by A through an iterative ZKP of its knowledge of the secret solution HC_t on G_t. In each iteration, S chooses a random permutation $\Pi_j(V_t)$ on the vertex set and uses it to build a graph $\Pi(G_t)$ isomorphic to G_t. The hash values $h(\Pi_j(V_t))$ and $h(\Pi_j(HC_t))$ are sent to A, who then chooses a bit b_j ($b_j \in_r \{0,1\}$). Depending on its value, S provides A with the image of the hamiltonian cycle through the isomorphism, or with the definition of the isomorphism. Once the authentication has been successfully carried out, A gives S the necessary information to have full access to protected resources.

Access Control Algorithm

Input: At stage r a supplicant node S that has been off-line from stage t wants to connect on-line to the network.

- $S \rightleftharpoons A$
- S sends openly $S \xrightarrow{o} A : G_t$
- A checks whether $t \leq r - T$
  - if $t \leq r - T$ then S is not authenticated
  - otherwise:
    - A and S agree $A \leftrightarrow S : l$
    - $\forall j \in \{1, 2, \ldots, l\}$
      1. S chooses $\Pi_j(V_t)$ and builds $\Pi_j(G_t)$ and $\Pi_j(V_t)$, isomorphic graph to G_t and corresponding hamiltonian cycle, respectively
      2. S sends openly $S \xrightarrow{o} A : \{h(\Pi_j(V_t)), h(\Pi_j(HC_t))\}$
      3. A chooses the challenge $b_j \in_r \{0,1\}$
      4. A sends openly the challenge $A \xrightarrow{o} S : b_j$
         (a) If $b_j = 0$ then S sends openly $S \xrightarrow{o} A : \{\Pi_j(G_t), \Pi_j(HC_t)\}$
         (b) If $b_j = 1$ then S sends openly $S \xrightarrow{o} A : \Pi_j$
      5. A verifies
         (a) that $\Pi_j(HC_t))$ is a valid hamiltonian cycle in $\Pi_j(G_t)$, if $b_j = 0$
         (b) that the hash function h on the result of Π_j on G_t produces $h(\Pi_j(G_t))$, if $b_j = 1$
    - if $\exists j \in \{1, 2, \ldots, l\}$ such that the verification is negative, then S is isolated.
    - otherwise A sends securely $A \xrightarrow{s} S$: the necessary information to have full access to protected resources of the network.

Output: Node S is connected on-line to the network.

2.5 Proofs of Life

All on-line legitimate nodes have to confirm their presence in an active way by broadcasting their proofs of life every certain interval of time lesser than T to all on-line legitimate nodes. If some insertion and/or deletion happens during such a period, a proof of life of every on-line legitimate nodes is distributed together with the broadcast necessary for those procedures. If no insertion or deletion happens during the period, the first node that has to prove its life executes the proof-of-life algorithm.

Proof-of-Life Algorithm

Input: At stage t node A has its $clock_A = T$, or it is executing step 4.(c) of the insertion algorithm or step 2 of the deletion algorithm.

1. A broadcasts $A \overset{b}{\leftrightarrow} network : A's\ proof\ of\ life$
2. – If A receives less than $n/2$ proofs of life as answers to her broadcast, she stops her proof of life and puts back her clock.
   – Otherwise: A broadcasts $A \overset{b}{\rightarrow} network : Received\ proofs\ of\ life$
3. All the nodes whose proof of life was broadcast initialize their $clock = 0$.

Output: At stage $t+1$ node A continues being an on-line legitimate node of the network.

2.6 Node Deletion

Each node must update its stored graph by deleting all nodes that have not sent any proof of life after a period T. Deletions are explicitly communicated to all on-line legitimate nodes so that any off-line node will be able to update its stored graph as soon as it gets access to the network.

Deletion Algorithm

Input: At stage t, A is a legitimate on-line node with $clock_A = T$, and v_i is a legitimate node, off-line for longer than T.

1. A checks the received proofs of life in A's FIFO queue, and detects those nodes $v_i \in V_t$ who have not sent any.
2. A broadcasts $A \overset{b}{\leftrightarrow} red : Possible\ deletion\ of\ v_i$
3. If A receives answers from more than $n/2$ nodes, A broadcasts $A \overset{b}{\rightarrow} red : Deletion\ of\ v_i$
4. All on-line nodes update $V_{t+1} = V_t \setminus \{v_i \in V_t \text{ with no proof }\}$
5. All on-line nodes update $E_{t+1} = E_t \setminus \{(v_i, v_j) : v_i \in V_t \text{ with no proof}, v_j \in N_{G_t(v_i)}\} \cup \{(v_j, v_k) : v_j, v_k \in N_{HC_t(v_i)}\}$
6. All on-line nodes update $HC_{t+1} = HC_t \setminus \{(v_j, v_i), (v_i, v_k)\} \cup (v_j, v_k) : v_i \in V_t$ with no proof, $v_j, v_k \in N_{HC_t(v_i)}$

Output: At stage $t+1$, node v_i has been completely deleted.

3 Simulation Analysis

The performance analysis of the proposal was done through the Network Simulator NS-2. Several Tcl based NS-2 scripts were created to produce output trace files, used both to do data processing and to visualize the simulation with Network Animator NAM. Also Tracegraph was used to analyze trace files, and setdest program was used to generate movement pattern files using the random waypoint algorithm. Many simulations were conducted of networks with different density grades (from 20 to 100 nodes), mobility rates (from 2 to 15 m./sec.), areas (from 400x400 to 800x800 m.), periods of simulation (from 60 to 200 sec.), event's rates (from 5% to 25%) and antenna ranges (from 100 to 250 m.). An example of simulation, an excerpt of the trace files and the associated hamiltonian cycles are respectively shown in Fig.1, Table 1 and Fig.2. Labels of nodes correspond to their different current states within the simulation:

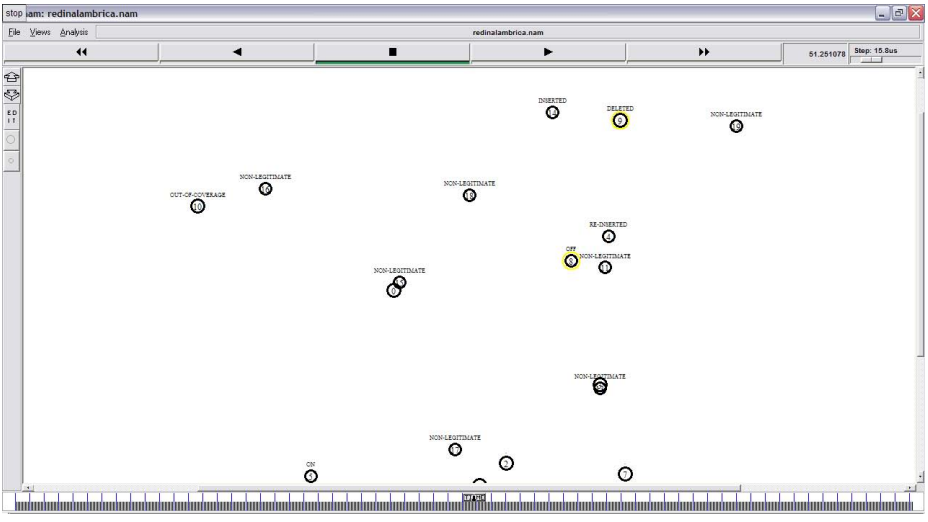


Fig. 1. Example of Network Simulation

Table 1. Example of Trace

Time	Event	H.C.
0.1	Nodes 0,1,2,3,4,5,6,7,8,9,10 are legitimate	10,2,6,8,1,5,9,4,3,7,0
1.10	Proof started by Node 5: Nodes 4,7 do not answer	
2.60	Node 7 meet Node 1: ZKP	
2.63	Node 9 switches off	
6.07	Proof started by Node 7: Node 9 do not answer	
7.20	Node 4 meets Node 9: ZKP	
9.88	Request of Node 18 to access: No node nearby	
14.52	Node 6 switches off	
16.40	Request of 14 to access: 10 starts Proof: 6 y 9 do not answer	10,2,6,8,1,5,9,14,4,3,7,0
23.24	Request of 11 to access: No node nearby	
25.30	Node 6 meets Node 1: ZKP	
32.59	Proof started by Node 0: Nodes 2, 6 y 9 do not answer	
34.08	Node 6 switches on: No node nearby: It switches off	
34.65	Request of Node 12 to access: No node nearby	
34.79	Node 2 meets Node 5: ZKP	
35.63	Off-line threshold is reached: Node 9 deleted	10,2,6,8,1,5,14,4,3,7,0
40.07	Proof started by Node 2: Not enough answers	
42.73	Node 5 switches off	
43.10	Node 6 meets Node 7: ZKP	
44.19	Node 5 switches on: ZKP with Node 2	
45.07	Proof started by Node 2: Nodes 10,6 do not answer	
45.97	Node 8 switches off	
59.96	Request of 16 to access: 1 starts Proof: 10, 6 y 8 do not answer	10,16,2,6,8,1,5,14,4,3,7,0

- Non-legitimate: Non-legitimate node inside the area of the network.
- On: Node that was previously off and now is trying to enter the network.
- Off: Node that has been off for a period shorter than T.
- Inserted: Node previously non-legitimate that has just been inserted.
- Deleted: Node previously off or out-of-coverage for longer than T.

– Re-inserted: Legitimate node previously off or out-of-coverage for shorter than T that has just been re-inserted.
– Out-of-coverage: Node that is out of network's coverage.

From the simulations we conclude that the protocol perfectly scales to any sort of networks with different levels of topology changes, and that a right choice of parameter T must be done according to number of nodes, bandwidth of connections, and computation and storing capacities of nodes.

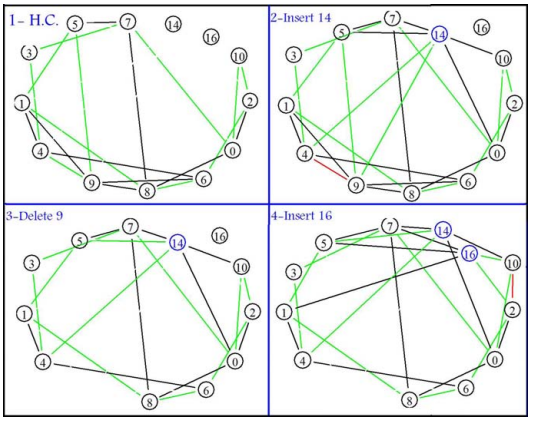


Fig. 2. Example of Graphs

4 Conclusions and Open Questions

This work describes a new strong authentication protocol specially designed for MANETs. The overall goal of the proposal has been to design a scheme that is able to react to network topology changes without the necessity of any centralized authority. Its core technique consists of a Zero-Knowledge Proof, which avoids the transference of any relevant information. The development of a full simulation of the proposal through the NS-2 network simulator has been carried out. The study of different applications, practical limitations and possible extensions of the proposed scheme may be considered open problems.

References

1. Aboudagga, N., Tamer, M., Eltoweissy, M., DaSilva, L., Quisquater, J.J.: Authentication protocols for ad hoc networks: Taxonomy and research issues. In: Proceedings of ACM Q2SWinet, ACM Press, New York (2005)
2. Asaeda, H., Rahman, M., Manshaei, H., Fukuzawa, Y.: Implementation of Group Member Authentication Protocol in Mobile Ad-hoc Networks. In: Proceedings of IEEE WCNC, Implementation of Group Member Authentication (2006)

3. Caballero-Gil, P., Hernández-Goya, C.: Strong solutions to the identification problem. In: Wang, J. (ed.) COCOON 2001. LNCS, vol. 2108, pp. 257–261. Springer, Heidelberg (2001)
4. Hahm, S., Jung, Y., Yi, S., Song, Y., Chong, I., Lim, K.: A Self-Organized Authentication Architecture in Mobile Ad-Hoc Networks. In: Meersman, R., Tari, Z. (eds.) On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE. LNCS, vol. 3291, pp. 689–696. Springer, Heidelberg (2004)
5. Saxena, N., Tsudik, G., Yi, J.H.: Efficient node admission for short-lived mobile ad hoc networks. In: Proceedings of IEEE ICNP, pp. 269–278. IEEE Computer Society Press, Los Alamitos (2005)
6. Weimerskirch, A.: Authentication in Ad-hoc and Sensor Networks. Ph.D. Thesis Ruhr-University Bochum. Germany (July 2004)
7. Wierzbicki, A., Zwierko, A., Kotulski, Z.: A New Authentication Protocol for Revocable Anonymity in Ad-Hoc Networks. In: Proceedings of IASTED CNIS (2005)

An Open Modular System for Monitoring Remote Sensors

Ignacio Solinis-Camalich, Alexis Quesada-Arencibia,
Jose Carlos Rodríguez-Rodríguez, and Roberto Moreno-Díaz jr

Instituto Universitario de Ciencias y Tecnologías Cibernéticas, Universidad de Las Palmas de Gran Canaria, E35017, Las Palmas, Spain
ignacio.solinis@gmail.com, aquesada@dis.ulpgc.es,
jcarlos@ciber.ulpgc.es, rmorenoj@dis.ulpgc.es

Abstract. The main goal of this project is to monitor and control remote sensors from one or more desalination plants. These plants can be found and distributed from each other miles away. All critical parameters must be controlled. In a desalination plant, the installed sensors are the key elements. These sensors feed the system providing the needed information so it could work properly. The plant state is defined by the sensors readings. From this state, we can identify possible anomalies in the normal behavior of the plant, which could cause the activation of different alarms. In order to accomplish these goals, a modular and flexible system has been developed not only for desalination plants but also for any other systems that require controlling remote sensors. The system allows direct access to this remote information through a variety of interfaces (web, SMS). The tier architectural design plays an important role in providing a basic characteristic: its adaptability to different environments. Therefore, adding or changing specific modules can be easily introduced into the modular structure of the system without interfering the core application modules. The application has been develop under Java technologies (J2EE) using Open Source Toolkits.

Keywords: Remote control, monitoring, sensor, alarm, video camera, SMS, J2EE, Struts, Eclipse, EJB3.

1 Introduction

The water in the Canary Islands is a precious and scarce good needed for the development of the archipelago. Desalination plants have been used for decades in the islands to process salt water, making it suitable for human consumption or for agriculture use.

Nowadays, there is a great need for controlling and monitoring systems remotely no matter it final location. In other words, to be able to access to the state of these systems no matter the time and its location (this is why it is called *remote control*[1]).

[1] A device or mechanism for controlling something from a distance.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 82 – 89, 2007.
© Springer-Verlag Berlin Heidelberg 2007

1.1 Final Objective

The purpose of the developed system is to offer a remote and controlled access to desalination plants. This access provides the state of the plant in real time from any location and at any time. The *state* is defined by the capture of lecture from key elements of the plant. These key elements are:

1. Sensor
2. Video camera
3. Alarms

We understand the concept of *lecture* as a set of data that define the state of a plant and that is needed so the system can work properly. The system obtains and recollects all the data to form a lecture. In this process, the basic element is the sensor reads obtained from the sensors connected to the plant.

The *alarms* are another identified key element associated with the state of the plants. These alarms are associated with sensors through the use of *activation rules*. These rules activate or deactivate alarms. For instance, an alarm activation will be produced by fulfiling an activation rule. This is why they are also known as validation rules. In short, the developed system allows to manage alarms associated with desalination plants (Fig. 1).

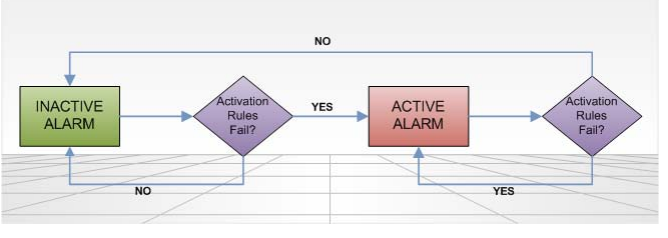


Fig. 1. This Data Flow Control Diagram (DFD) defines and controls the process of activation and deactivation of alarms

1.2 System Structure

The developed system can be divided into two different subsystems, one for the desalination plants and another one for the office. Both subsystems have to interact so the system can work properly. Therefore, what we finally have is a centralized network of desalination plants, whose central node is the office subsystem (Fig. 2).

The office subsystem is in charge of managing centralized data related to desalination plants. The plant subsystem manages everything about a plant. The Office manages and coordinates tasks as a central node. Once a plant has been added correctly to the Office, two communication schemes can be establish (this is a bidirectional communication).

1. *Scheme 1* (office-to-plant): In this scheme, the office starts the communication process and query the plants to get the state of the plant in real time

a. The data sensor's reads
b. The list of active alarms
c. The list of images capture by the video cameras

2. *Scheme 2* (plant-to-office): The plant sent messages warnings to the office to alert about activation and deactivation of alarms

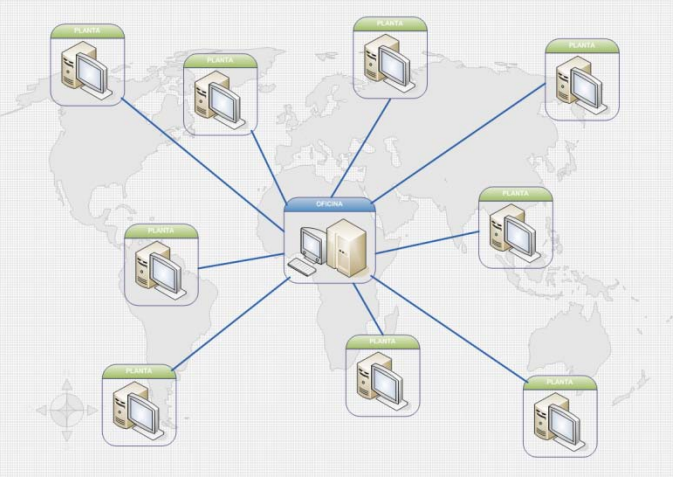


Fig. 2. Topology of the system. You can see in this picture both subsystems in a centralized topology where desalination plants acts as *peripheral nodes* and the office as the *central node.*

1.3 Development with Open Source Toolkit

One of the main objectives in this project was the use of open source toolkits for the development process. This can be resumed in

- A full development with Java and associated technologies (J2EE)
- Use of open source software
  - Operative System (Linux)
  - Specific software (application server, databases, etc.)
  - Integrated development environment (Eclipse, DBDesigner, etc.)

One of the main advantages of using open source software is the reducing development costs. This is due basically to the saving in proprietary software use. Despite, there is a big community of developers behind open source software projects that can be very helpful during development process.

Another advantage is the use of J2EE and Java, because you make the application platform-independent thanks to the use of a virtual machine like JVM (the Java Virtual Machine).

2 System Architecture

In this section, we are going to introduce briefly the system architecture. We are going to talk about its modularity and the modules interactions. Therefore, we are going to present some of the fundamental modules define in the system.

We have to consider that the most important part of the system is the one involving the plant. Here is where we have the *core of the system*. The main objective of the core is to obtain and process the lectures. This is why we are going to talk on this section only about the plant architecture.

2.1 Modular Architecture of a Desalination Plant

The functionality required by the system, and obtain during the analysis, has been divided into different modules. The database associated and managed by each module and the interaction with each other has been identified and defined in the architectural design. Here is a brief introduction to the most relevant modules of the plant's architectural design (Fig. 3).

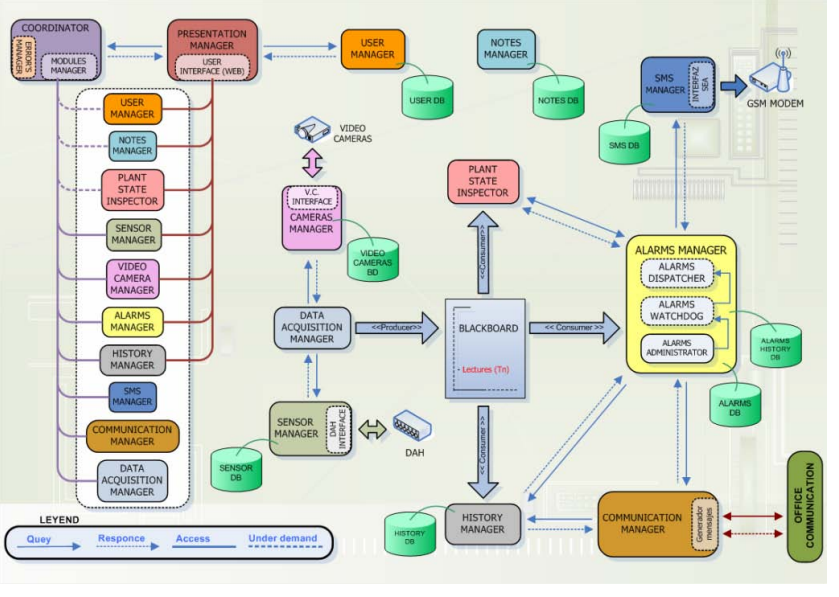


Fig. 3. Modular architecture. We can see in this picture the different modules of the plant architecture, the interaction with each other and their access to their specific databases.

Core of the System. Within the architecture of the plant, a group of modules works and interact very closely together forming the *core of the system*. These are the modules involving the production and consumption of data readings from the plant. We will talk about these modules in the following subsections.

Data Acquisition Manager. This is the central module of the system, and it is part of the *core of the system*. The main goal of this module is to get and form all the lectures[2]. Once a lecture has been formed, it is forward to any module willing to receive it. This scheme can be identified as a *Producer/Consumer* concurrent

[2] We understand the concept of *lecture* as a set of data that define the state of the plant.

Table 1. Producer/Consumer concurrent example, involved modules

PRODUCER	CONSUMER
Data Acquisition Manager	History Manger
	Alarms Manager
	Plant State Inspector

example. Here we have a module acting as a *Producer* of messages, and some other modules acting as *Consumers* of these messages (Table 1). A message is what we have defined as a lecture. These messages are written into a shared area called "*the Blackboard*". Here is where the reads and writes are made by the corresponding modules.

In short, the *Data Acquisition Manager* is in charge of forming and writing the lectures into the Blackboard. In order to be able to form a lecture, this module has to interact with other two modules. These modules are:

1. *Sensor Manager*: This module has a direct communication with the sensors, getting its readings in real time.
2. *Video Camera Manager*: This module gets of images capture by the video camera in real time.

Once the Data Acquisition Manager obtains both sensors and video camera reads, this module forms and writes the lecture into the Blackboard. Any other module can read these lectures from the Blackboard. There are some key elements related to this process.

1. *System's clock*: A lecture comes with a timestamp that marks it. This mark associated with the lecture defines the exact moment at which the lecture was formed. This timestamp identifies a lecture throughout the application. The system's clock is used to get the timestamp. Therefore, a timestamp cannot be shared by more than one lecture (as an identifier).
2. *Base refresh rate*: This defines the writing rate of lectures into the Blackboard. Therefore, synchronization between readings and writings into the Blackboard is needed. This synchronization is being made by the base refresh rate. Therefore, this refresh rate affects both producers and consumers of messages equally.

Sensor Manager. This module, as we have mentioned before, gets the sensor reads and forward it to de Data Acquisition Manager so this module can form and produce the final lecture. These sensors are connected to a device called Data Acquisition Hardware (DAH). Therefore, this module gets the sensor reads through the DAH once a sensor has been added correctly to the system. A sensor is added by registering it configuration data to the system, so the system can access it correctly. This module access and manages the database relating to sensors.

Data Acquisition Hardware (DAH). There is a physical device needed to access the sensors connected to the desalination plant. This device is called as Data Acquisition Hardware (DAH). The DAH carries out the signals from the sensors to the final

computer. Then these signals can be process by the computer in order to get the readings in real time. The Modbus Protocol[3] is being used to manage the communication between the computer and the DAH. A DAH's interface is being added to the Sensor Manager so this module can access sensor readings in real time. Therefore, we can change the communication to the sensors by only change this interface.

Video Camera Manager. The images capture by the video cameras are being obtained by this module. This is possible thanks to an SDK from Pleora Technology. An interface to the video cameras has been added to this module. This module access and manages the database relating to video cameras. A Video Camera has to be added and register to the system in order to be able to capture images from then.

Alarms Manager. This is one of the most important modules. It is used to manage the alarms in the system. To make this process correctly, the functionality has been divided into three sub modules. These three sub modules need to interact with each other to be able to reach their goals. These are the modules:

1. *Alarms administrator*: This module manages and administrates anything related to alarms from adding to modifying or deleting. It has complete access to the alarms database so it can manages then correctly.
2. *Alarms Watchdog*: This module makes the detection of alarms in the system. More precisely, this module detects activation and deactivation of alarms. Once an activation or deactivation has been detected, this module has to inform to the alarms dispatcher.
3. *Alarms dispatcher*: This module is used to inform to other modules about activations and deactivations of alarms. Some of these modules can be very critical, like the SMS Manager (SMS, Short Message Service).

SMS Manager. This module receives message warning about activation and deactivation of alarms from the Alarms Manager. Then, this module has to form a SMS message and send it to the right mobile phones. This process can be critical and very important if the alarms warning have a high priority (defining the importance of the warning).

3 Implementation

The system has been implemented under a J2EE platform. J2EE is a platform-independent, Java-centric environment from Sun for developing, building and deploying Web-based enterprise applications online. The J2EE platform consists of a set of services, APIs, and protocols that provide the functionality for developing multitier Web-based applications.

We finally have implemented the Blackboard with the Java Message Service, which provides an asynchronous communication system (Fig. 4).

[3] Modbus Protocol is a messaging structure developed by Modicon in 1979. It is used to establish master-slave/client-server communication between intelligent devices.

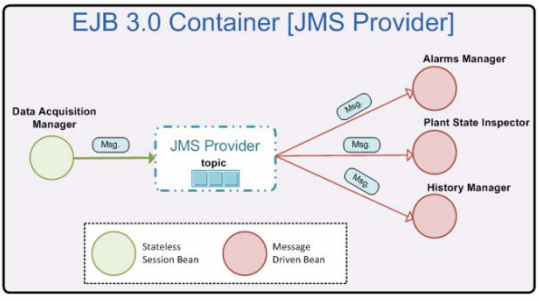


Fig. 4. This is an example of how the Blackboard was finally implemented under J2EE

We have used JBoss 4.0.5 GA as the application server where the final system is running. For the web tier (presentation tier) we have used Struts. EJB 3.0 specification for the component tier. Finally, MySQL as the Database that stores all the necessary data so the system can works properly. For sending SMS messages, we have used SMSLib. SMSLib is a Java library which can be used to send and receive SMS messages via compatible GSM modems.

4 Conclusions and Future Works

The control and maintenance of a desalination plant is done normally in situ. A technician has to go to the plant and check everything, so the only way of discovering any possible failure in the plant is during this kind of visit, and sometimes these failures had a great impact over the plant's behavior. In conclusion, the use of a remote control application can really improve the maintenance of a desalination plant reducing the associated costs.

One of the possible future works could be the introduction of an Actuator Module. Right now, the system cannot interact with the plant's devices. For instance, this module could stop the plant after receiving an SMS message, or it could change some parameters in the behavior of some process.

Another possible future work, quite related with the Actuator Module, could be the introduction of an Expert System. An expert system can help in the diagnosis process to discover possible anomalies or future failures in the plant. Despite, with the Actuator Module, it could not only help in the diagnosis process but also interact with the plant in an autonomous way.

In conclusion, a modular and flexible system has been developed not only for desalination plants but for any other systems that require controlling remote sensors. The system allows direct access to this remote information through a variety of interfaces (web, SMS). The tier architectural design plays an important role in providing a basic characteristic: its adaptability to different environments. Therefore, adding or changing specific modules can be easily introduced into the modular structure of the system without interfering the core application modules.

References

1. Cockburn, A.: Writing Effective Use Cases. Addison-Wesley, London, UK (2001)
2. Bell, J.T.: J2EE Open Source Toolkit. Building an enterprise platform with open source tools. Wiley, Chichester (2003)
3. Alur, D.: Core J2EE Patterns. Best practices and design Strategies, 2nd edn. Prentice Hall, Englewood Cliffs (2003)
4. Cavaness, C.: Programming Jakarta Struts. O'Reilly (2002)
5. Sriganesh, R.P.: Mastering Enterprise JavaBeans 3.0. Wiley, Chichester
6. Keith, M.: Pro EJB3. Java Persistance API. Apress

A Structure for Generalized Space-Time Processing: The Feedforward Visual Case

R. Moreno-Díaz[1], A. Moreno-Díaz[2], G. De Blasio[1], and D. Freire-Obregón[1]

[1] Institute for Cybernetics, Universidad de Las Palmas de Gran Canaria, Spain
rmoreno@ciber.ulpgc.es, gdeblasio@dis.ulpgc.es, dfreire.ciber@gmail.com
[2] School of Computer Sciences, Universidad Politécnica de Madrid, Spain
amoreno@fi.upm.es

1 Formal Frame

Traditional interpretation of early visual image processing in a retinal level has focused exclusively on spatial aspects of receptive fields (RFs). We have learned recently that RFs are spatiotemporal entities and this characterization is crucial in understanding and modelling circuits in early visual processing ([2]). We present a generalization of the layered computation concept to describe visual information processing in the joint space-time domain.

The starting point is the generalization of the concept of layered computation ([3], [5]). In its original sense, a modular system is structured in layers of similar computing elements. Modules in each layer operate on some input space to provide an output space. Both spaces have a pure physical nature such that input and output lines represent denditric trees, axons or, simply, wires. Outputs can interact with inputs through any type of feedback.

In the generalization, we preserve the existence of layers of computing elements which perform, in general, any type of decision rule or algorithm over both input and output spaces. But the nature of these spaces is changed to be spaces of representation. They are multidimensional spaces where each axis represents possible values of an independent variable which measures an independent possible property of inputs and outputs.

In the formal structure, the input space of the generalized computational frame consists of a spatio-temporal cube or hypercube in which one axis is time and the others correspond to cartesian coordinates (two or three dimensions) or to symbolic space representations. Notice that what it is needed in the input space representation is the possibility to define *distances* as proximities in that space: events close in the space correspond to events close in the property time space.

Operations on the input space act as a generalized convolution over a volume to produce, at each instant and each output point, an output layer to contribute to the output space. In the general case, volumes in the output space are also taken into account by every computational module to decide what the local value of the output layer should be. That is, there is feedback and also, cooperative processing since the outputs of nearby computing units in previous instants are

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 90–96, 2007.
© Springer-Verlag Berlin Heidelberg 2007

used to compute the present output of the unit, due to the space-time character of the feedback sample.

2 The Feedforward Visual Case

In the visual case with no feedback the input space comprises a stack of slides. The computing units perform convolution-like operations on space-time volumes, which are the spatio-temporal overlapping receptive fields.

In general, operations performed on the spatio-temporal receptive field of a computational element can be arbitrary (algorithmic, algebraic, probabilistic, etc.). What is important in this representation is that no previous outputs can be taken into account to generate a new output slide. That is, no recursive process can be realized by the computing structure. But it is still somehow cooperative as it happens to the visual system of vertebrates, in the sense that the same input data to one unit is shared by all units in the proximities. Cooperativity, as it is well known, is a key factor in reliable layered computation ([4]).

3 Computation and Tool Description

As it corresponds to the visual level with no feedback, we have developed a visual information processing system following the general principle above. It is capable of performing spatio-temporal operations of an arbitrary nature on a stack of images or a video stream. Computing units are pixels in each image and their receptive fields may assume a variety of configurations. Examples are presented which illustrate various classical filtering space-time effects, linear and non-linear. The novelty is the implementation of the classical neurophysiological concept of *center-periphery* but now in space-time, which produces naturally contrast detection, edge and movement detection, ON, OFF and ON-OFF behaviour, directionality sensitivity and others.

3.1 Generalized Convolution

The tool allow us to process a set of images using a spatio-temporal structure. Each and every image is filtered and the filter mask is a 3D-geometric figure centered, each time, in the image and pixel being processed. Operations implemented involve calculations with the center and periphery of each pixel's receptive field, including linear as well as non linear operations (threshold). Next section explains how these regions (center and periphery of each pixel) can be properly selected.

3.2 Receptive Field Geometry

Selecting those regions acting as center and periphery of each pixel's receptive field is easily performed by the Tool through a series of guided steps. This

selection concerns not only the regions themselves but also their shapes and properties.

Figure 1 shows the input of the system, a stack of images to be processed. This stack of images is arranged in a temporal sequence, each one is an evolution of the previous one.

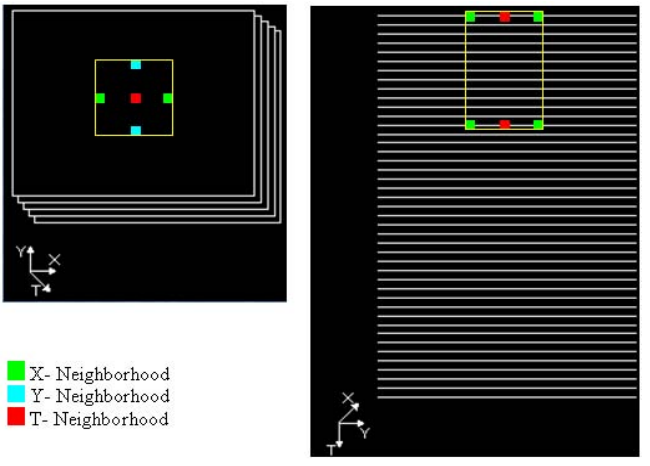


Fig. 1. Selection of strategical points

In this stack of images, different selection processes are carried out: first, a primary image in the stack to begin the process with, second, spatial neighbourhoods (axis X and Y) for a generic pixel and third, the depth parameter (axis T) or number of neighbourhood images taking part in each image processing. There is one output image for every input image. Each output image is the result of a filtering operation involving each pixel, its spatial neighbours and its temporal neighbours.

These neighbourhood selections define the receptive field geometry for each pixel and they are made to form two geometrical models: hexahedron or ellipsoid. Appropriate setting of its parameters, corresponding to X, Y and T axis respectively, will generate a wide variety of geometrical figures: cubes, prisms, spheroids, etc. Inside each figure, two geometric volumes define center and peryphery. For instance, Figure 2 shows a symmetric sphere configuration for center and peryphery. Altough named center, it has to be neither in the center nor to conform a symmetric whole with the peryphery itself: a wide variety of configurations, including asymmetric ones, are possible.

4 Results

To illustrate the functioning of this tool we show results for different contrast detection tests in a purely spatial way, in a purely temporal way and in a

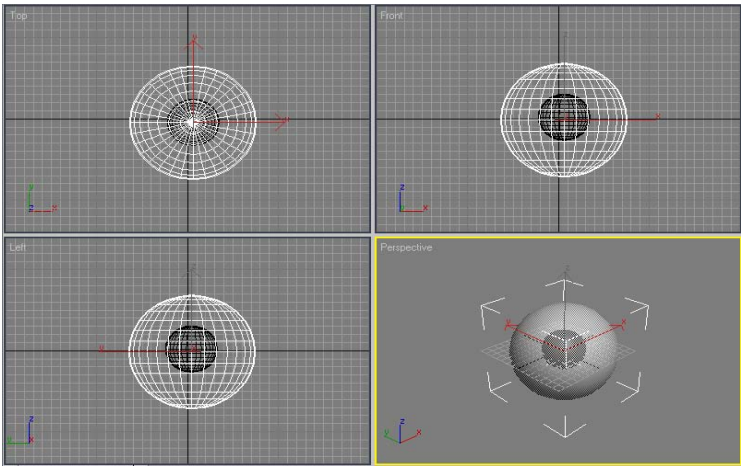


Fig. 2. A Generalized Convolution Center-Periphery forming geometric figures

spatio-temporal way. The tool is implemented in Matlab 7.0. Filter masks used in each experiment are shown in Figure 3. The highlighted pixel in each mask acts as the center and its surround as the peryphery. In the convolution process center pixels are weighed positively and peryphery pixels are weighed negatively. In a space-time coordinate frame with X, Y and T axes, these filters correspond to hexahedral figures having coordinates $3 \times 3 \times 1$, $1 \times 1 \times 3$ and $3 \times 3 \times 3$ respectively.

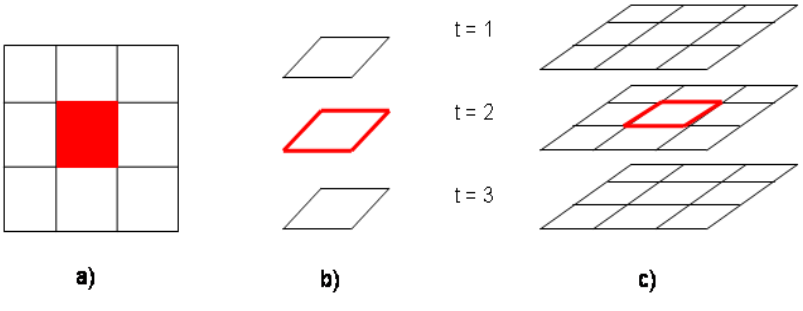


Fig. 3. Filter Masks

The sequence of input images for the experiments is shown in Figure 4. This sequence shows an object moving from left to right. Each image corresponds to a different instant in time, that is, first one is for $t = 1$, second for $t = 2$ and so on.

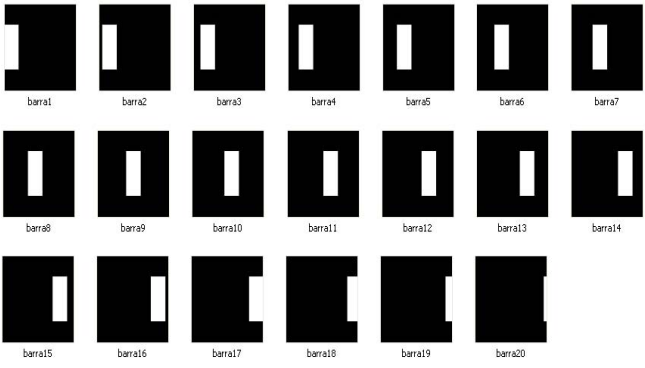


Fig. 4. Input set of images

Filtering the sequence with the spatial filter mask shown in Figure 3a) produces the output shown in Figure 5. It is a spatial contrast detector. Each slide in the output captures the object contour and position in a static way at each time instant. Before the filtering process starts, the tool inserts white images for $t = 0$ right before the first one provided by the user. That's the reason for the first white image shown in each output sequence.

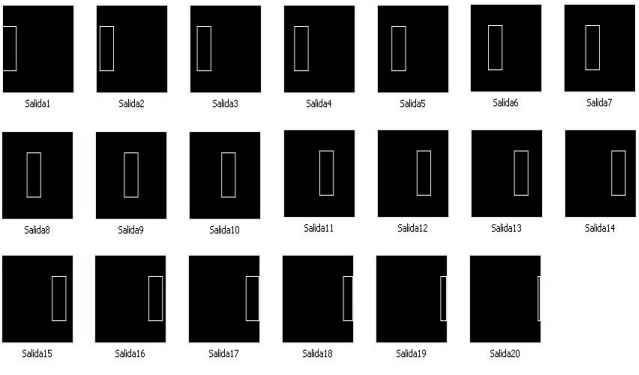


Fig. 5. Spatial Contrast Detector

Figure 6 shows the output using the filter mask in Figure 3b). It's a temporal contrast detector. Each slide in the output captures object's movement, distance covered (left white bar) and distance to be covered (right white bar).

Finally, Figure 7 shows the spatio-temporal contrast detector using filter mask in Figure 3c). Each slide in the output captures object's contour, its edges and current position, as well as object's movement, distance covered and distance to be covered in the next period of time.

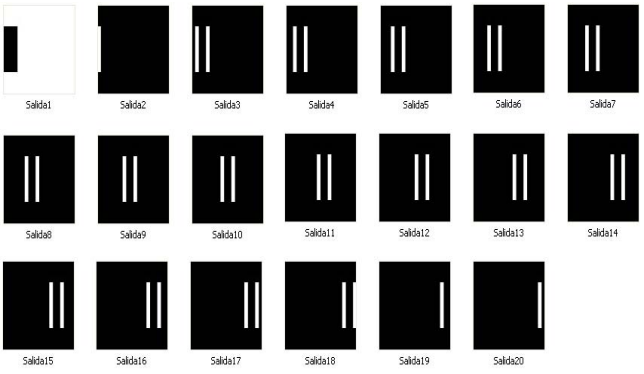


Fig. 6. Temporal Contrast Detector

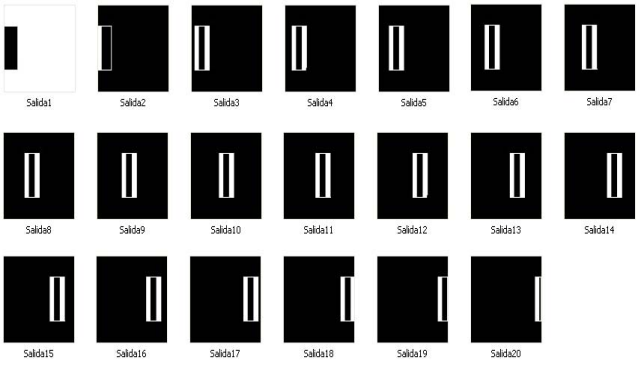


Fig. 7. Spatio-Temporal Edge Detector

5 Conclusions

Although this project is still in its initial steps, we have shown results concerning the application of the simplest linear filters detecting contrast and movement, implementing a spatio-temporal version of the neurophysiological concept of center-peryphery. The convolution process described is based on linear spatio-temporal summation which holds reasonably well for most simple cells ([1]). Each output slide is the result of a processing involving images in different time instants as well as different locations in each image. Chromatic processing is another aspect yet to be considered. The ultimate goal is the implementation of filters emulating the sequential processing of visual information in different biological retinal models such as rabitt's or frog's.

We are currently working on tool's speed optimization. We have implemented an interface Matlab 7.0 - C++, improving eficiency and processing time. The

tool has a friendly interface and its design minimizes the number of mouse clicks to start the filtering process. It has also a zoom facility making easier and more precise the selection of important areas around each pixel.

References

1. DeAngelis, G.C., Ohzawa, I., Freeman, R.D.: Spatiotemporal Organization of Simple-Cell Receptive Fields in the Cat's Striate Cortex. II. Linearity of Temporal and Spatial Summation. J. Neurophysiol. 69, 1118–1135 (1993)
2. DeAngelis, G.C., Ohzawa, I., Freeman, R.D.: Receptive-field Dynamics in the Central Visual Pathways. Trends Neurosci. 18, 451–458 (1995)
3. McCulloch, W.S.: Embodiments of Mind. The M.I.T. Press, Cambridge, Massachusetts (1965)
4. Mira, J., Delgado, A., Moreno-Díaz, R.: Cooperative Processes in Cerebral Dynamic. In: Lainiotis, D.G., Tzannes (eds.) Applications of Information and Control Systems, D. Reidel Pub. Comp., pp. 273–280 (1980)
5. Moreno-Díaz, R., Rubio, E.: A Theoretical Model for Layered Visual Processing. Int. J. Bio-Med. Comp. 10, 134–143 (1979)

Arrays and Continuous Attributes

Margaret Miró-Julià

Departament de Ciències Matemàtiques i Informàtica
Universitat de les Illes Balears
07122 Palma de Mallorca, Spain
margaret.miro@uib.es

Abstract. Computers store and exploit knowledge, at least that is one of the aims of Artificial Intelligence and Information Systems research fields. However, the problem is to understand what knowledge means, to find ways of representing knowledge, and to extract useful information from stored data. Scientific and engineering disciplines are developing daily. This progress is strongly connected to complex techniques and methods. However, it is surprising to observe that the majority of data tables (Information Systems) considered have binary attributes, and therefore discrete attributes values, and of course, the methods and techniques used are binary based. This paper presents an approach for handling data tables with multiple valued continuous attributes by considering an unusual algebra. The proposed algebra does not handle raw data, it handles declarative descriptions of the data.

1 Introduction

Changing data into knowledge is not a straightforward task. Data is generally disorganized, contains useless details and may be incomplete. Knowledge is the opposite, organized but expressed using a poorer language, it might be imprecise or vague.

Before proceeding, let's take a minute to analyze the subtle and elusive difference between binary and multivalued descriptions. Given a set of m objects $D = \{d_1, d_2, \ldots, d_m\}$ and a set of n binary attributes $R = \{r_1, r_2, \ldots, r_n\}$ a binary Object Attribute Table (OAT) can describe a situation. Subsets of D are described in terms of the binary entries of the n attributes. A subset of D may be described by a function $f : \{0,1\}^n \to \{0,1\}$. Declarative expressions for a function of a binary data table can be obtained by means of a Boolean algebra. A Boolean algebra is a convenient tool when considering attributes that take binary values. Can an equivalent algebra that handles non binary continuous attributes be obtained?

2 Discrete Algebra

Recently, a non binary algebra that allows the treatment of multiple valued data tables (information systems) with discrete attributes has been introduced [1]. In

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 97–104, 2007.
© Springer-Verlag Berlin Heidelberg 2007

order to develop a technique suited for multivalued discrete data, the following had to be created:

- A symbolic representation of subsets was developed. This new representation allows the consideration of subsets of discrete attribute values. Set $\mathcal{S}_c = \{c_1, c_2, \ldots, c_k\}$ is formed by all possible symbols describing subsets of C using the octal code [2]. So that the expression $c_i \looparrowright C_i$ may be read as "c_i is the symbol describing subset C_i".
- The concept of universe was explored. The introduction of the universe allows the consideration of sets of attributes. Therefore, the theory allows sets of attributes that take subsets of discrete attribute values. Special subuniverses called cartesian and co-cartesian subuniverses were introduced and studied [3].
- The theory of subuniverses together with the subset's symbolic representation allow the introduction of the concept of array and co-arrays by means of the cross product.
  An array is a description of subsets of 3-spec-sets (chain ordered description of 3 specifications) that can be written as a cross product $|t_i| = |c_i, b_i, a_i| \looparrowright C_i \otimes B_i \otimes A_i$ where $c_i \looparrowright C_i$, $b_i \looparrowright B_i$ and $a_i \looparrowright A_i$. Arrays with two discrete attributes can be represented graphically as shown in Figure 1.

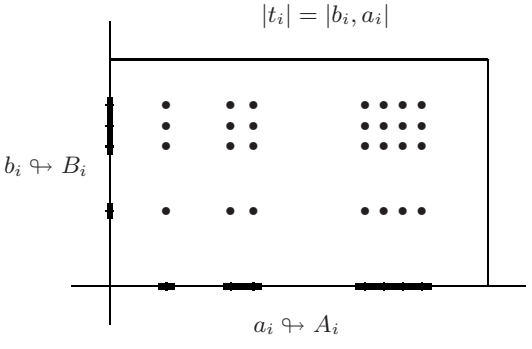


Fig. 1. Array with discrete attributes

  A co-array is a description of subsets of 3-spec-sets (chain ordered description of 3 specifications) whose complement can be written as a cross product $||t_p|| = ||c_p, b_p, a_p|| \looparrowright \sim (\hat{C_p} \otimes \hat{B_p} \otimes \hat{A_p})$ where $c_p \looparrowright C_p$, $b_p \looparrowright B_p$ and $a_p \looparrowright A_p$. Co-arrays with two discrete attributes can be represented graphically as shown in Figure 2. An algebra based on the idea of arrays and co-arrays was developed. This algebra provides a symbolic way to declaratively describe subuniverses in terms of an expression [4].
- In the same way that one may have equivalent declarative descriptions of the same subuniverse, one may have equivalent expressions. The possibility of determining some identifiable canonical expressions were explored. In this

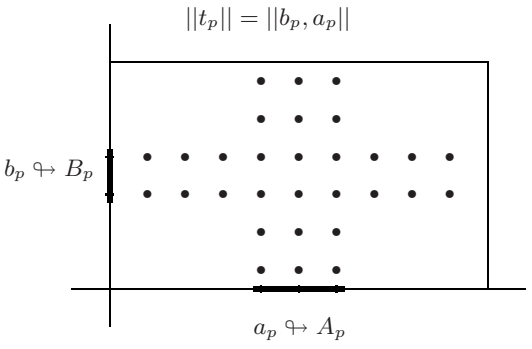


Fig. 2. Co-arrays with discrete attributes

endeavor, prime-ars and prime-co-ars were defined and a prime-ar algorithm and prime-co-ar algorithm constructed [5].
– Since the number of prime-ars can be uncomfortably large, a simpler prime-ar expression, called cover, was introduced. A declarative approach to determine a satisfactory cover was offered [6].

3 Continuous Algebra

This paper addresses a parallel algebra that deals with continuous attribute values. This algebra does not handle raw data, it handles declarative descriptions of the multivalued continuous data and allows a multivalued continuous description of the knowledge contained in a data table by means of array expressions. Hence, this paper proposes the introduction of:

1. a symbolic representation of subsets of continuous attribute values of a given attribute by the introduction of disjoint intervals of the attribute.
2. the continuous universe and continuous subuniverses analogous to the discrete ones, that will consider more that one continuous attribute.
3. the basic elements of the continuous algebra that provide a description of the special continuous subuniverses. That is, the introduction of arrays and co-arrays for continuous attribute values by means of a continuous cross product.

4 Subsets of Continuous Attribute Values

Definition 1. *Let* $D = \{d_1, d_2, \ldots, d_i, \ldots, d_m\}$ *be an ordered set, called domain, of elements* d_i *representing the* m *objects, let* $R = \{r_g, \ldots, r_c, \ldots, r_a\}$ *be a set of the* g *attributes or properties of the objects. The set of continuous values of attribute* r_c *is represented by set* I^C *formed by disjoint intervals* $I^C = \{[I_n^c], ..., [I_2^c], [I_1^c]\}$. *The elements of set* I^C, $[I_j^c]$, *are called 1-spec-intervals*

since the elements are defined by means of one continuous specification. A Multivalued Object Continuous Attribute Table (OCAT) or Multivalued Continuous Information System is a table whose rows represent the objects, and whose columns represent the continous attributes of these objects. Each element $[I_j^c]$ *represents the interval (continuous value) of attribute* r_c *that corresponds to object* d_i*. Table 1 illustrates a Multivalued Object Continuous Attribute Table.*

Table 1. Object Continuous Attribute Table

	r_g	...	r_c	...	r_a
d_1	$[I_1^g]$	...	$[I_1^c]$	...	$[I_1^a]$
d_2	$[I_2^g]$	...	$[I_2^c]$	...	$[I_2^a]$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
d_i	$[I_i^g]$	...	$[I_i^c]$	...	$[I_i^a]$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
d_m	$[I_m^g]$	...	$[I_m^c]$	...	$[I_m^a]$

In order to handle the Multivalued Object Continuous Attribute Table, where attributes values are given as intervals from a given set of continuous values, a new multivalued algebra is proposed. The starting point is a multivalued language that describes all possible subsets of intervals.

4.1 Subsets of Intervals

Let's specify in more detail what exactly is set I^C. Attribute r_c takes continuous values, the range of all possible values is $C = [c_{min}, c_{max}]$. This range is divided in continuous intervals $[I_j^c]$, j = 1, 2, n, such that $[I_1^c] \cup [I_2^c] \cup \ldots \cup [I_n^c] = [c_{min}, c_{max}]$ and $[I_i^c] \cap [I_j^c] = \emptyset$, $\forall i, j$. One way of satisfying these properties is considering intervals closed on the left and opened on the right, except the last interval that is closed on both sides. Throughout this paper $[I_j^c]$ is an interval closed on the left and opened on the right. Set I^C is formed by all such intervals $[I_j^c]$.

It has already been mentioned that $[I_j^c]$ is an element of I^C, but if several distinct elements of I^C are considered, a subset of I^C can be defined. Let I^{C_j} be such a subset, $I^{C_j} = \{[I_i^c] \mid [I_i^c] \in I^C\}$, that is, $I^{C_j} \subseteq I^C$.

If set I^C along with all possible subsets is considered the introduction of regular set operations, union ($\cup$), intersection ($\cap$) and complement respect to I^C ($\widehat{\ }$), generates a lattice.

4.2 Multivalued Language

If a symbolic representation or a description of subsets of intervals is considered, there is a parallel lattice $< \mathcal{S}_c, +, \cdot, \widehat{\ }, I^0, I^1 >$ defined on the set $\mathcal{S}_c$ of all possible

symbols representing subsets of I^C. The zero of this algebra is I^0 (the symbol representing the empty set). The identity is I^1 (the symbol representing set I^C).

Throughout this paper, the expression $c_j \looparrowright I^{C_j}$ may be read as "c_j is the symbol describing the subset of intervals I^{C_j}". The symbolic representations of regular set operations complement ($\widehat{}$), union ($\cup$) and intersection ($\cap$) are:

$$\hat{c_h} \looparrowright \widehat{I_h^c} \qquad c_h + c_k \looparrowright I^{C_h} \cup I^{C_k} \qquad c_h \cdot c_k \looparrowright I^{C_h} \cap I^{C_k}$$

The generic symbolic representation and operations for discrete attribute values have been carefully studied in [3].

5 Continuous Universe and Continuous Subuniverse

All the concepts and operations introduced above make reference to only attribute, that is, only one set I^C. A multivalued OCAT has more than one attribute (more than one column).

Let $R = \{r_c, r_b, r_a\}$ be a set of 3 attributes whose attribute values are given by the sets of intervals $I^C = \{[I_n^c], ..., [I_2^c], [I_1^c]\}$, $I^B = \{[I_n^b], ..., [I_2^b], [I_1^b]\}$ and $I^A = \{[I_n^a], ..., [I_2^a], [I_1^a]\}$. The elements of I^C, I^B, I^A are 1-spec-intervals (one specification intervals). A 3-spec-interval, $[I_k^c, I_j^b, I_i^a]$, is a chain ordered description of 3 specifications, one from set of intervals I^C, one from set of intervals I^B and one from set of intervals I^A.

The 3-spec-intervals brings to mind the ordered triples that form the cartesian product. But the cartesian product has an undesirable property:

$$A \times B \neq B \times A.$$

Therefore, to avoid this unwanted characteristic, each spec-interval represents itself and all possible permutations.

Definition 2. *The cross product $I^G \otimes \cdots \otimes I^B \otimes I^A$ is the set of all possible g-spec-intervals formed by one interval of I^G, ..., one interval of I^B and one interval of I^A.*

$$I^G \otimes \cdots \otimes I_j^b \otimes I^A = \{[I_x^g, \ldots, I_j^b, I_i^a] \mid [I_x^g] \in I^G, \ldots, [I_j^b] \in I_j^b, [I_i^a] \in I^A\}$$

Once again, it is important to mention that the cross product is not the cartesian product. A g-spec-set represents itself and all possible permutations whereas the elements of the cartesian product are different if the order in which there are written varies. There is a need to determine an order in a g-spec-set. The basis T is an ordered chain $< I^G, \ldots, I^B, I^A > \equiv T$ which establishes the sequential order in which the spec-sets are always written.

The set of all possible g-spec-intervals induced by the sets of intervals $I^G, \ldots, I^B$, I^A is called the **continuous universe** and every subset of the universe is called a **continuous subuniverse**.

6 Continuous Arrays and Co-arrays

Definition 3. *Given g continuous attributes, let* $I^{G_i} \subseteq I^G$, ..., $I^{B_i} \subseteq I^B$, $I^{A_i} \subseteq I^A$, *an array* $|t_i| = |g_i, \ldots, b_i, a_i|$ *is the symbolic representation of the cross product* $I^{G_i} \otimes \ldots \otimes I^{B_i} \otimes I^{A_i}$ *where* $g_i \looparrowright I^{G_i}$, ..., $b_i \looparrowright I^{B_i}$, *and* $a_i \looparrowright I^{A_i}$.

$$|t_i| = |g_i, \ldots, b_i, a_i| \looparrowright I^{G_i} \otimes \ldots \otimes I^{B_i} \otimes I^{A_i}$$

Definition 4. *Given g continuous attributes, let* $I^{G_p} \subseteq I^G$, ..., $I^{B_p} \subseteq I^B$, $I^{A_p} \subseteq I^A$, *the symbolic representation of the complement (in the continuous universe) of the cross product of subsets* $\widehat{I^{G_p}} \otimes \ldots \otimes \widehat{I^{B_p}} \otimes \widehat{I^{A_p}}$ *where* $g_p \looparrowright I^{G_p}$, ..., $b_p \looparrowright I^{B_p}$, *and* $a_p \looparrowright I^{A_p}$ *is called a co-array.*

$$||t_p|| = ||g_p, \ldots, b_p, a_p|| \looparrowright \ \sim (\widehat{I^{G_p}} \otimes \cdots \otimes \widehat{I^{B_p}} \otimes \widehat{I^{A_p}})$$

Arrays and co-arrays with two continuous attributes can be represented graphically in a manner similar to the discrete case, as is shown in Figures 3 and 4.

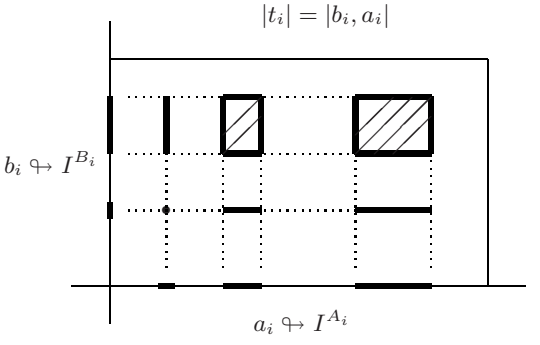


Fig. 3. Array with continuous attributes

7 Expressions

Since the arrays describe subuniverses (subsets of spec-intervals), regular set operations may be performed with them.

Let $|t_i| = |g_i, \ldots, b_i, a_i|$ and $|t_j| = |g_j, \ldots, b_j, a_j|$ be two continuous arrays, the following operations are introduced:

– $\sim$ complement (symbolic representation of the complement of a subuniverse respect to the universe):

$$\sim |t_i| \looparrowright \ \ \sim (I^{G_i} \otimes \ldots \otimes I^{B_i} \otimes I^{A_i})$$

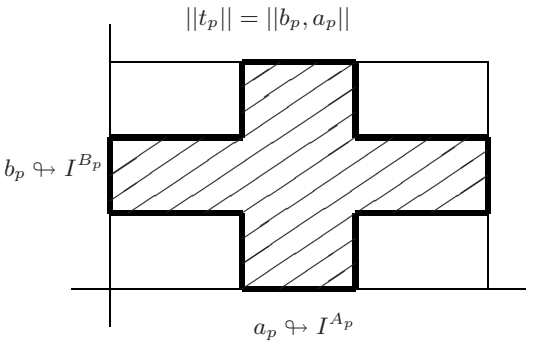


Fig. 4. Co-array with continuous attributes

– ‡ sum (symbolic representation of the union of two subuniverses):

$$|t_i| \ddagger |t_j| \looparrowright (I^{G_i} \otimes \ldots \otimes I^{B_i} \otimes I^{A_i}) \cup (I^{G_j} \otimes \ldots \otimes I^{B_j} \otimes I^{A_j})$$

– ∘ product (symbolic representation of the intersection of two subuniverses):

$$|t_i| \circ |t_j| \looparrowright (I^{G_i} \otimes \ldots \otimes I^{B_i} \otimes I^{A_i}) \cap (I^{G_j} \otimes \ldots \otimes I^{B_j} \otimes I^{A_j})$$

The ∘ product of two continuous arrays is a continuous array.

All the results obtained by use of operations ∼, ‡ and ∘ on arrays are symbolic representations of subuniverses. If only two attributes are considered, these operations can be represented graphically as shown in Fig. 5.

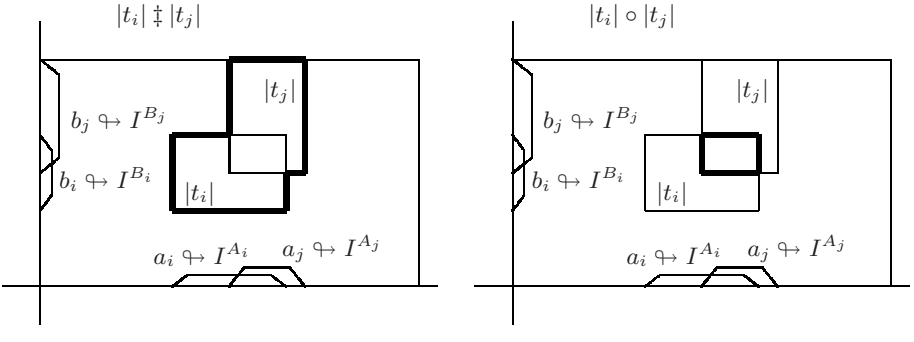


Fig. 5. 2-dimensional ‡ sum and ∘ product of arrays

Definition 5. *Every combination of arrays using operations* ∼, ‡ *and* ∘ *(well formed formula) is called an expression* E_i.

$$E_i = \sim |t_i| \ddagger |t_j| \circ |t_k| \ldots$$

Definition 6. *An expression E_i is called an array expression if it is written as a ‡ sum of arrays:* $E_i = |t_z| \ddagger \cdots \ddagger |t_y| \ddagger \cdots \ddagger |t_x|$.

Continuous subuniverses can be described by algebraic expressions of continuous arrays. A continuous expression is a symbolic representation of a continuous subuniverse. In general, an expression represents a partial reality included in a multivalued OCAT. An Object Attribute Table with continuous attributes can be described by an array expression of continuous arrays.

8 Conclusion and Future Work

A multivalued language that deals with continuous data has been introduced. The symbols that constitute this language must form a lattice. This continuous language allows for the introduction of continuous arrays and co-arrays similar to the discrete arrays and co-arrays. An Object Attribute Table with continuous attributes can be described by an expression formed with continuous arrays and continuous co-arrays.

The proposed continuous array algebra does not handle raw data, it handles continuous declarative descriptions of the data. All continuous multivalued Object Attribute Table can be written as an continuous array expression

Finally, it should be mentioned that this array algebra has a dual version, the co-array algebra that has been introduced in [3] and should be further studied.

Acknowledgements

This work has been partially supported by the Dirección General de Investigación del Ministerio de Educación, Ciencia y Tecnología through the TIN2004-07926 project.

References

1. Miró-Julià, M., Fiol-Roig, G.: An algebra for the treatment of multivalued information systems. In: Perales, F.J., Campilho, A., Pérez, N., Sanfeliu, A. (eds.) IbPRIA 2003. LNCS, vol. 2652, pp. 556–563. Springer, Heidelberg (2003)
2. Miró-Julià, M.: A New Approach for Handling Information Systems. In: Isaias, P., Kommers, P., McPherson, M. (eds.) Proceedings of the IADIS Intenational Conference. e-Society, vol. 1, pp. 549–556 (2004)
3. Miró-Julià, M.: A Contribution to Multivalued Systems. PhD thesis, Universitat de les Illes Balears (2000)
4. Miró, J., Miró-Julià, M.: A Numerical Computation for Declarative Expressions. In: Moreno-Diaz, R., Pichler, F. (eds.) EUROCAST 1997. LNCS, vol. 1333, pp. 236–251. Springer, Heidelberg (1997)
5. Miró-Julià, M., Miró, J.: Transformation of Array Expressions. In: Proceedings of the Second IASTED International Conference. Artificial Intelligence and Applications, pp. 273–278 (2002)
6. Miró-Julià, M.: A Minimal Cover for Declarative Expressions. Frontiers in Artificial Intelligence 100, 375–386 (2003)

Neuro-Immune-Endocrine (NIE) Models for Emergency Services Interoperatibility

Zenon Chaczko[1] and Perez Moses[2]

[1] Faculty of Engineering, University of Technology Sydney, Broadway, NSW 2007, Australia
zenon@eng.uts.edu.au
[2] ZInd, 16 Gregory Terrace Lapstone, 2773, Australia
zind@aapt.net.au

Abstract. Highly dynamic, re-configurable hardware, embedded software and communication networks are becoming very significant in operation of various emergency services. The key challenge in designing such systems is to provide a framework for interoperability of emergency services in disaster situations. The goal of the paper is to provide an insight into modelling techniques for studying emergency services interoperability functions in system design to avoid hidden points of failures. Concepts of artificial Neuro-Immune-Endocrine (NIE) homeostatic models [21][22][24] for autonomous self-configuring and self-healing systems are discussed. The paper features examples of collaborative software agents' behaviour in hostile environments, cooperating protocols, smart embedded devices and pro-active infrastructures in various areas related to emergency services operations.

1 Introduction

This research paper examines how biology inspired design methodologies can be applied to the design of highly adaptable, large-scale and context (i.e. emergency situation or/and hostile environment) aware software infrastructure. Our ultimate goal is to provide a robust and scalable design interoperability models of infrastructure software that can handle systems containing virtually thousands of embedded devices, sensors and actuators spanning large areas and cities as well as manufacturing plants, building complexes and infrastructure facilities that can collaborate to detect and analyse complex situations and unexpected events. The work addresses various issues involved in modelling, design and algorithmic work involved in implementation of such software intensive systems combining best software engineering practices and innovative approaches to modelling of software system architecture that draw inspiration from biological teleonic processes that involve complex interoperating networks of genes, proteins, cells and biological organs. The paper discusses various ideas for using biological analogies to perform required adaptations. These include low-level agent-cell attention, agent autonomy as well as plasticity of the brain/nervous, immune and endocrine regulatory functions of the network-based super system

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 105–112, 2007.
© Springer-Verlag Berlin Heidelberg 2007

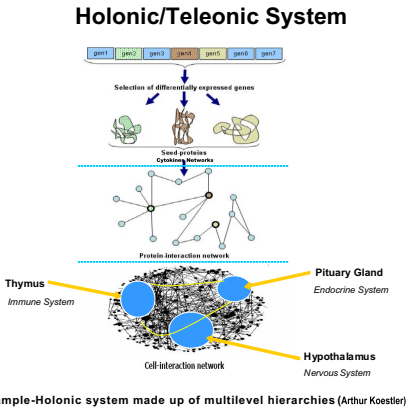


Fig. 1. Network based Supersystem

(see the below Fig 1) The concepts of organic computing takes inspiration from the way biological organisms are able to adapt to a complex dynamic environment. Rather than to exactly specify every single state and response the idea is to: (i) sketch a rough outline of the system containing plausible possibilities for key states and responses, and (ii) provide a set of rules that allow the system to autonomously adapt and evolve the appropriate states and responses over the course of its lifetime. As obvious as the above might seem, in particular since it has been functioning for biological systems for millions of years, today it is far from clear where, how, and if at all this idea can be applied to technical systems. This work presents the results of an initial, conceptual architecture study that aims to answer the above question. To this end, we look at a specific class of systems that has recently generated much interest: large-scale environment aware sensor networks and discuss (i) the problems involved in implementing such systems using conventional engineering approaches, (ii) a system architecture that can circumvent these problems by evolving a system optimised for a given application domain from a rough generic template, and (ii) specific initial ideas for using biological analogies to perform the required adaptation.

This paper outlines the introduction of the biology inspired, hybrid multi-agent based concepts and methods for modelling, simulating and architecting collaborative activities of humans and computer subsystems in emergency service organisations that can be referred to as policies and work practice modelling. We introduce the ontology and describe an experimentation framework for architecting, simulating and re-design of policies and work practices emergency services. We describe the model and its applicability in context of service interoperability. The problem spaces entails: researches and comparison on the manufacturing processes and application, resource allocation and planning, risk resource and analysis, design and application of smart agents for infrastructure oriented software systems Our research into modeling and design methods for

emergency service oriented processes and applications involves adaptation of Holonic Technology Platform (HTP) [15][18] [19] for the design of biomimetic and holonic middleware as well as design of bio-inspired algorithms for the middleware components (services) Focus is to adapt Artificial Nervous-Immune-Endocrine System (NIE) paradigms and algorithms the work involves design and development of concrete applications in Safety and Security of Emergency Services as well as Resource Management Allocation and Planning domains. The main goals and motivation for our work are as follows: (1) Provide effective modelling/design techniques for studying emergency services interoperability. (2) Adopt biology inspired NIE models/algorithms for autonomics self i.e. perimeter control, intrusion detection, sensory, (3) Create a Holonic framework [16][17] and a set of tools to support of operations emergencies services Fire Brigades, Police, Ambulance Services and (4) The ultimate motivation: Interoperability among emergency agencies around the globe. The research approach involves several phases such as: (1) Holonic/Teleonic Knowledge Base for infrastructure oriented holonic software, (2) Extension of the HTP by biology inspired agents components/constraints, (3) Simulation of disasters/emergency situations in virtual environment, (4) Modeling and implementing of system (and agents) behavior/collaboration between groups; and (5)Development of Middleware for building interoperating systems and devices for emergency services. Since the project involves translation of operational modules based on NIE models (NN, AIS, AES) this invariably requires a use of some form of Knowledge Base to encapsulate in holons, agents and biomimetic ontologies (NS, IS, ES): policies, work practices and rules, workflows and tasks, objectives and constraints, agents and holons.

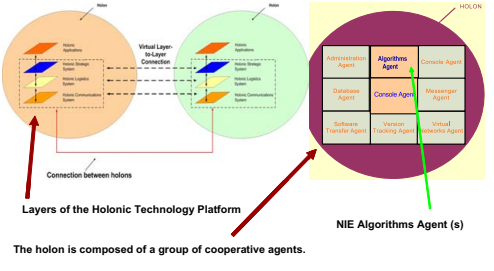


Fig. 2. HTP Architecture (courtesy of [15])

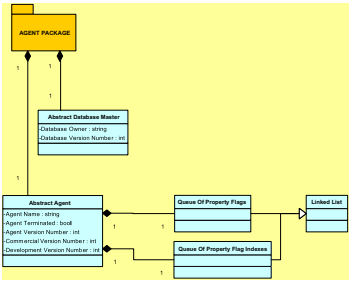


Fig. 3. The Holon (courtesy of [15])

2 Overview of HTP

An agent (including an abstract *agent - software agent*) is an construct that can represent many items [5]. An *agent* possess a name (identifier) and a location. The *agents* can form a representation framework. *Agents* embody their own goals (i.e. BDI), *agents* have a local perspective on the world. *Agents* interact with each

other and with their environment [20]. The Holonic Technology Platform (HTP) [14,15,18,19] is a framework with a holonic architecture (Fig2) facilitates the development of distributed intelligence systems [9] for both wireline and wireless environments.The HTP consists of the Holonic Communications System (HCS), the Holonic Logistic System (HLS) and the Holonic Strategic System (HSS). In a wireless environment, the HCS is not needed. Both HLS and HSS packages include agent architectures [15] peer-to-peer technologies [7][8] and holonic intelligence. The HTP is developed in Java using its platform independent, multithreaded and support for distributed [7] [8] [10] environment capabilities. Java provides packages that can easily manipulate code. In the HTP-based Holonic Management System (HMS) scenario the Fire Brigade Agent (FBA) on Node A has been conversing with Paramedic Agent (PMA) on Node D. After some time, the FBA is transferred to Node D. (where the distance FBA-Node A is >> Node A- Node D) The new traffic between nodes is now be reduced since the node has been transferred. There are two types of algorithms in the HTP: System algorithms (application wide) and Agent algorithms (can be embeded in a specific holon see Fig 3.) System algorithms affect the functionality of the system, i.e., when an agent on a holon is doing too much work, a system algorithm may decide to transfer the agent to the participating holons so that the system workload is more balanced. Agent algorithms affect the functionality of an agent, i.e., when a group of agents are working on a set of tasks, an algorithm may be initiated to determine the order of performance.

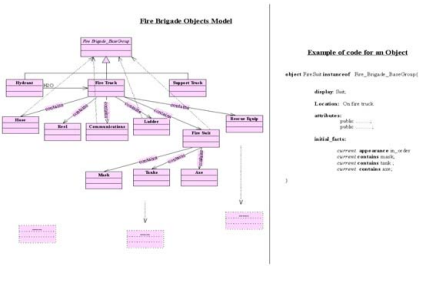


Fig. 4. Fire Brigade Object Model

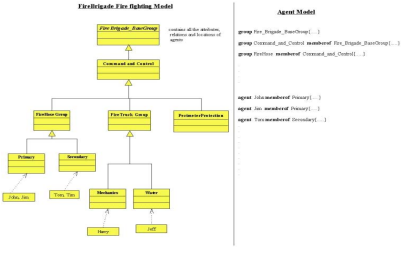


Fig. 5. Fire Brigade Agent Model

Construction of agents typically involves defining: (1) Agent Id and possibly its relations to other agents-groups (i,e. represents the actor, agent, and how they relate to each other.), (2) Agent activity (i.e what type of activities does the group and or agent perform), (3) Agent communication (i.e what the agent communicate with other agents or objects, or objects to objects as seen in the Fig 5 and Fig. 6.), (4) Agent's activity timing/constraints (i.e.prior to conducting and activity the agent might check if a particular condition or constraint is fulfilled in order to complete the event, constraints could be physio-biological nature, health, level of training; inthe HTP we call it a thoughtframe), (5) Knowledge representation (i.e what is the initial state of the agent, what does it know (facts)

and beliefs-*doxa-greek trans.* does it possess about the world.), (6) Objects: (i.e we need to describe all the objects/items used by agents in order to conduct the specific activity), (7) Environment (i.e geographical environment in which agents and objects perform their activities, areas and travel paths [12]).

3 Case Study: Emergency Services Security System

Emergency Services Security System (E3S) environment: high level of unscheduled (chaotic) and ad-hoc activities, strict policies, issues with efficient use of resources: systems, people and materials. Emergencies require: predicting failure/risk assessment, resource management i.e.Scheduling: people, services, operations, activities, networks,tools and etc. In multi-disciplinary domain: Road traffic management/Transportation Telecommunications Specialised Equipment The system has a hierarchical (holonic) structure yet dynamic nature of events requires a flexible approach to the organisation. The E3S unit (holon/agent /cell) has to choose the appropriate strategy itself, the unit has capacities for autonomy and co-operation. In the problem space it has to be able to assertain types of security threats: accidents, malicious intrusions [11] and blocking. Typical applications that use immnocomputing models to combat threats are: Intrusion detection systems (IDS), Perimeter Control System, Antivirus software (AVS) [27][28]. There are many issues with these applications as they could be resource demanding,use incompatible interfaces, reliant on human intervention, cannot respond to new strains of threats and, new type of emergency situations. The E3S solution is required to: quickly detect the intrusion, develop/use countermeasures (remedies), nullify (eliminate) threat and distribute an "antidotum". The main E3S requirements are listed in the Table 1and the main system components are shown in Fig. 9. Autonomous processes should result in streamlining. Perimeter intrusion detection functions and localisation [2] is required to use generic detectors and be able to monitor TCP/IP packets of data. the immunocomputing algorithm uses r-contiguous bit matching rule (for example, B-cell Detector [1 * * * 0 * 1 * 1 0 1 * * * * * * 1 1 1 0 * * * ... * 0 1 0 * * *) Experimental Design involved To demonstrate r-contiguous bit matching rule is viable and advantageous over brute force and other methods of string matching. Demonstrate matching capabilities of r-contiguous approach using shorter/less detectors. Demonstrate threshold can be adjusted to increase detection range of a set of detectors. Demonstrate performance gains using r-contiguous approach and its role in sampling.

4 Final Remarks

Our research contributions resulted in construction of a network security system framework using holonic components Work involved construction of a distributed multi-layered (holonic) systemwith resource pooling mechnism. The system requires further automation of processes to minimise user input (simplicity) The immunocomputing algorithms embeded in agents in the system deal with false

Table 1. SS2 Requirements List [13]

Problem	Solution/Requirements
Resource Demanding	Distribute load; share resources; lightweight.
Require human operator	Autonomy through learning and well defined processes
Can not respond to new straihs of attacks or emergencies	Learning mechanism allowing the system to self organise methods of detection and remedies
High false positives	Multilayered approach, different detection methods, each communicating with each other to keep themselves in the loop.
Property	**Paradigms Applied**
Detectors	T-cell, B -cell and memory cell.
Immune Principle	Self and non -self, Major Histocompatability Complex, negative selection, and affinity maturation.

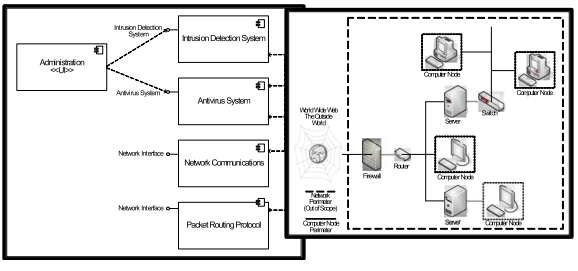
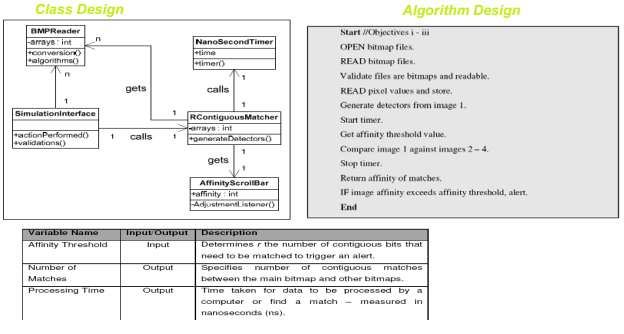


Fig. 6. E3S Components [13]



Variable Name	Input/Output	Description
Affinity Threshold	Input	Determines *r* the number of contiguous bits that need to be matched to trigger an alert.
Number of Matches	Output	Specifies number of contiguous matches between the main bitmap and other bitmaps.
Processing Time	Output	Time taken for data to be processed by a computer or find a match – measured in nanoseconds (ns).

Fig. 7. E3S R-Contiguous Matching Rule Simulation [13]

positives at large,use feedback loops, provide a global analysis of events and use both T-cell and B-cell mechansim. Through our system development work and biology inspired models, we discover more about the elements of cognition and learning as well as their boundaries In the long run there is no full guarantee that our approach would lead to a major breakthrough, yet a very real potential does exist. The process of building these systems, as with many scientific pursuits,

will inevitably point toward answers to some fundamental questions concerning human nature itself. Further enhancements will include broader, more flexible detector coverage. In implementations of AIS algorithms within HTP framework the self and non-self mechanism can be replaced by a new model of the danger theory [1]. Further research will aim to explore Artificial Endocrine model and homeostatic algorithms

References

[1] Aickelin, U., Cayzer, S.: The Danger Theory and Its Application to Artificial Immune Systems, Technical Report Information Infrastructure Laboratory, HP Laboratories Bristol, HPL-2002-244 4th of (September, 2002) (Accessed May 2006), http://www.hpl.hp.com/techreports/2002/HPL-2002-244.pdf

[2] Chaczko, Z., Klempous, R., Nikodem, J., Nikodem, M.: Methods of Sensors Localization in Wireless Sensor Networks. In: ECBS 2007. 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, Tucson, pp. 145–152 (2007)

[3] Concurrency Accessed (December 2006), http://java.sun.com/docs/books/tutorial/essential/concurrency

[4] Cougaar Accessed (December 2006), http://www.cougaar.org

[5] Definition of Agent Accessed (December 2006), http://www.ryerson.ca/dgrimsha/courses/cps720/agentDef.html

[6] Gong, L.: Project JXTA: A Technology Overview (October 2002)

[7] Helsinger, A., Thome, M., Wright, T.: Cougaar, A.: Cougaar: A Scalable, Distributed ulti-Agent Architecture. In: IEEE International Conference on Systems, Man, and Cybernetics, The Hague, Netherlands, vol. 2, pp. 1910–1917 (2004)

[8] Itao, T., Tanaka, S., Suda, T., Aoyama, T.: A Framework for Adaptive UbiComp Applications Based on the Jack-In-The-Net Architecture. ACM Wireless Network J. 10(3), 287–299 (2004)

[9] Jablonski, A., Klempous, R., Licznerski, B.: Diversified Approach to Methdology and Technology in Distributed Intelligent Building Systems. In: Moreno-Díaz Jr., R., Pichler, F. (eds.) EUROCAST 2003. LNCS, vol. 2809, pp. 174–184. Springer, Heidelberg (2003)

[10] JADE Agents, Accessed (June 2006), http://www.ryerson.ca/~dgrimsha/courses/cps720/JADEFIPA.html

[11] Klempous, R., Nikodem, J., Radosz, L., Raus, N.: Adaptive Misbehavior Detection in Wireless Sensors Network Based on Local Community Agreement. In: ECBS 2007. 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, Tucson, pp. 153–160 (2007)

[12] Kulbacki, M., Jablonski, B., Klempous, R., Segen, J.: Learning from Examples and Comparing Models of Human Motion. JACIII 8(5), 477–481 (2004)

[13] Lam, J.T.Q.: Biological Immune System Inspired Computer Network Security System Theoretical Framework: Comparative Study of Immune System, Capstone Report, UTS, Sydney (21 November, 2006)

[14] Ng, C., Alibhai, Z., Sabaz, D., Uncu, O., Gruver, W.A.: Framework for Developing Distributed Systems in a Peer-to-Peer Environment. In: IEEE International Conference on Systems, Man, and Cybernetics, Taipei, Taiwan (2006)

[15] Ng, C.: Framework for Developing Distributed Systems in a Peer-to-Peer Environment, M.ASc. thesis defence presentation v.7 Simon Fraser University, Vancouver, Canada (December 2006)

[16] Ovcharenko, S., Alibhai, Z., Ng, C., Sabaz, D., Gruver, W.A.: Implementation of a Wireless Distributed Intelligent System. In: IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications, Prague, Czech Republic (2006)
[17] Ostaszewski, M., Seredynski, F., Bouvry, P.: Immune Anomaly Detection Enhanced with Evolutionary Paradigms. Published in the proceedings of the conference on Genetic and evolutionary computation (GECCO) (July 8-12, 2006)
[18] Sabaz, D., Gruver, W.A.: Distributed Intelligent Systems: What Makes Them 'Intelligent'. In: IEEE Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications Proceedings, Beijing, China (2005)
[19] Sabaz, D.: The Slow Death of Centralization: The Ascension of a Distributed World, Simon Fraser University's IRMACS Presentation, Vancouver, Canada (2006)
[20] Snyder, R., Tomlinson, R.T.: Robustness Infrastructure for Multi-Agent Systems. In: Proceedings of the Open Cougaar, New York, U.S.A (2004)
[21] Traversat, B., et al.: Project JXTA 2.0 Super-Peer Virtual Network (May 2003)
[22] Vargas, P., Moioli, R., de Castro, L.: Artificial Homeostatic System: A Novel Approach. In: Capcarrère, M.S., Freitas, A.A., Bentley, P.J., Johnson, C.G., Timmis, J. (eds.) ECAL 2005. LNCS (LNAI), vol. 3630, pp. 754–764. Springer, Heidelberg (2005)
[23] Watkins, A., Timmis, J., Boggess, L.: Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Machine Learning Algorithm. Genetic Programming and Evolvable Machines Journal 5(3), 291–317 (2004)
[24] Wierzchon, S.T.: Deriving concise description of non-self patterns in an artificial immune system. In: Jain, L.C., Kacprzyk, J. (eds.) New Learning Paradigm in Soft Computing, pp. 438–458. Physica-Verlag (2001)
[25] Wu, Z., Dong, H., Liang, Y., McKay, R.I.: A Chromosome-based Evaluation Model for Computer Defense Immune Systems. In: Proceedings of the IEEE Congress on Evolutionary Computation, Canberra, Australia, pp. 1363–1369 (December 8-12, 2003)

Automatic Change Detection in Dynamical System with Chaos Based on Model, Fractal Dimension and Recurrence Plot

Mateusz Tykierko

Institute of Computer Engineering,Control and Robotics
Wroclaw University of Technology
ul. Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
Mateusz.Tykierko@pwr.wroc.pl

Abstract. Automatic change detection is the important subject in dynamical systems. There are known techniques for linear and some techniques for nonlinear systems, but merely few of them concern deterministic chaos. This paper presents automatic change detection technique for dynamical systems with chaos based on three different approaches neural network model, fractional dimension and recurrence plot. Control charts are used as a tool for automatic change detection. We consider the dynamical system described by the univariate time series. We assume that change parameters are unknown and the change could be either slight or drastic. Methods are checked by using small data set and stream data.

1 Introduction

The main goal in many signal processing or system identification tasks is detection of the change in dynamical systems, based on limited information. In this paper the available information is defined by one dimensional time series formed from measurements of the system output.

There are well established change detection methods for liner and some for nonlinear systems, but there is lack of methods considering chaotic systems. Chaos in mathematical sense is a strong irregular behavior, which may seems to be random but it is not. A dynamical system is classified to chaotic group based on its properties. The non-integer dimension, the sensitivity to initial condition, the positive Lyapunov exponent are some of them [15].

The change detection in a dynamical system with chaos has little attention in literature. Articles [8,4] consider change detection technique in Lorenz and Mackey-Glass system where the change is introduced by modification in the differential equation parameters. Those papers consider drastic changes in the parameter value which causes the fact that the system is no longer chaotic and slight changes which were 20% of original parameter value. The EEG signal change detection is presented in [7].

Chaotic phenomena occur in a vast research area: astronomy, biology, social science, cryptology, quantom physics and many more. As the example of the

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 113–120, 2007.
© Springer-Verlag Berlin Heidelberg 2007

chaotic behavior we can named: engine combustion chamber, laser led, hear arrhythmia, computer network, electronic circuit, disease spreading.

Change detection in dynamical systems could be viewed as prediagnostic element. The detected change is a stimulus or warning, which may, protect from breakdown or catastrophe, reduce the production stoppage, save lives, protect environment or help in a cryptographic analysis.

In this paper we focus on the change detection in chaotic systems where the values of the parameter change are unknown. The range of the change value varies from slight changes which do not influence the system chaotic behavior to drastic changes which makes a transition from chaos to a point or a cyclic attractor. We focus on the small changes, since in our opinion small changes are more challenging. The proposed methods are based on Radial Basis Function network model [5], estimation of fractal dimension [18] and recurrence plot [9]. The automatic detection is realized by control charts.

2 Change Detection

Our task is to find an unknown change which appears at an unknown time in a chaotic system with access to a univariate time series, formed from a system output. The change detection method should be characterized by following properties: minimal change detection time, minimal number of false alarms, short computation time, minimal number of method's parameters, minimal resource overhead. We placed emphasis on minimizing the above properties during method description.

We specify following origins of change in dynamical systems: the input signal change; the system dynamics change; the noise type or the noise level change. In this paper we only focus on the system dynamics change. Some issues of the change detection in the presence of noise are also presented but they are in the context of the system dynamics change.

Proposed method of the automatic change detection is the combination of two steps. The first step is the monitoring of a certain quantity the second takes advantage of statistical control method in order to detect changes automatically. We study three different approaches to monitoring step: the RBF neural network model, the fractal dimension and the recurrence rate.

A typical control chart is a graphical display of a quality characteristic that has been measured or computed from a sample versus a new data. Control limits are chosen so that if the process is in control, nearly all of the sample points will fall between them. As long as the points plot within the control limits, the process is assumed to be in control and no action is necessary. But when one of this limits is crossed, the process is assumed to be out of control and an alarm is raised. In our terminology alarm is equal to the presence of a change. For detailed discussion on quality control and control charts we refer to [13]. We use the Exponentially Weighted Moving Average (EWMA) control chart. It is a statistic for monitoring the process that averages the data in a way that gives less and less weight to data as they are further removed in time.

The system evolution in time is only known through the univariate time series, thus the system dynamics have to be reconstructed.

One of the most interesting thing in the chaotic systems is that almost every dynamical quantity carries information about system dynamics such as Lyapunov exponents, attractor dimension, eigenvalues of fixed points etc. Thus, measuring only one of them could be enough to conclude from the system dynamics.

A phase space reconstruction from the time series is the next step in system analysis. This can be achieved by using delay coordinate maps. Transformation which does not change the properties of the attractor is called embedding. It is proved, [17], that delay coordinate map is a diffeomorphism and in a certain condition is an embedding. Key parameters of the coordinate map are embedding dimension and embedding delay.

The choice of the embedding parameters should be driven by the time series properties. The false nearest neighbors method [11] is usually used for finding the embedding dimension. In [3] it is suggested that the optimum embedding delay is at the first minimum of the mutual information. When the phase space is reconstructed the next step towards change detection is obtaining the quantities which points to a change.

2.1 Example of Change Detection

The Lorenz system are utilized to check the quality of the RBF network, the fractal dimension and recurrence plot as a change detector. It is well studied in literature and often used to test new methodologies. The system is chaotic. The chaotic behavior implies non-integer dimension of the system attractor. A positive Lyapunov exponent makes the system unpredictable on the long term scale.

Two kinds of changes, slight and drastic was introduced to Lorenz system parameter. After the drastic change, the system quickly converges to the point attractor. The further experiments will be based only on the slight change since the drastic change is easily detectable by the standard statistical methods.

Radial Basis Network Model. In the model based change detection approach the model accuracy is monitored over the time. The model prediction quality is crucial and influences the accuracy and time of change detection. In order to detect a change in time series the RBF neural network model was used. Prediction in the chaotic time series based on RBF is presented in following papers [6,16].

We used the heuristic called forward selection [14] to build the RBF network. It starts from an empty subset, in every iteration add one hidden unit until given criterion stops decreasing. This approach searches for the subset with minimal prediction error in a discrete space of subsets of a set of hidden units with fixed centers and width.

The state space reconstruction procedures are used to construct the training set. The centers of the sets of candidates are located in elements of training set.

Gaussian function with fixed length was selected. We use small regularization value due to fact that the proper modeling of the chaotic time series is impossible without regularization [6]. For detail procedure of calculating the weights, the width and the regularization of the RBF network we refer the reader to [5,14].

The builded model was tested for the prediction quality and proper dynamic reconstruction. Example of prediction in Lorenz system is depicted in Fig. 1. The second figure shows the fast divergence of the prediction value due to chaotic properties of the system. The correlation dimension and the maximal Lyapunov exponent was calculated on predicted series and they were in agreement with value given in literature [6].

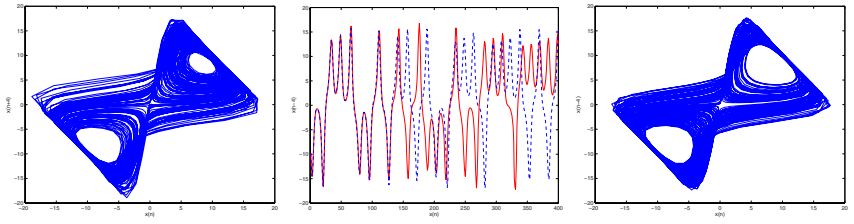


Fig. 1. First figure - the Lorenz attractor, second figure - the original time series solid line and its prediction doted line, third figure the Lorenz attractor predicted by the network

The model was initialized and the difference between the network output and the system output was monitored. Significant difference indicates a change. To avoid the fast divergence of the prediction the model was restarted with current value of system output after n iterations. Thus the difference is calculated on non overlapping windows of predicted values. In Fig. 2 the change detection of slight changes is depicted. Large changes above 10% of original parameter value can be easily detected using one step predictor. In order to detect whole change region difference has to be calculated on a window with fixed length. The small changes are hard to detect. Changes of 1% of the original parameter value was not detected correctly. In case of 2% system parameters change the detection is possible but it is not so obvious as in previous change case. One step predictor detects the change but the error value in the change free region is comparable with the values in the change region, which leads to the problem with fixing the threshold. If the threshold would be to small, there will be the false alarms. However, using the difference calculated on window the change is well defined and can be easily detectable by the control chart. Disadvantage of this approach is requirement of building and testing a model.

Fractal Dimension. The fractal dimension is a quantity that gives an indication of how completely a fractal appears to fill the space. The dimension has a number of related definitions [18]. We use the correlation dimension for most

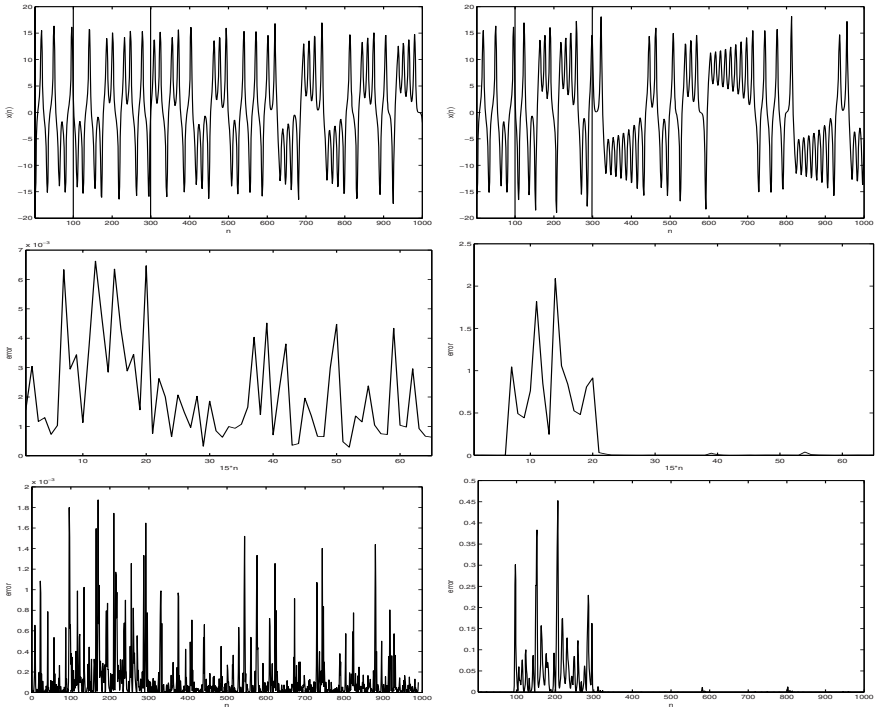


Fig. 2. First column: the time series with 2% change between step 100-200; the error between the system and the model added on window length 15; the error in each step. Second column: the time series with 20% change between step 100-200; the error between the system and the model added on window length 15; the error in each step.

of our experiments. The correlation dimension has the advantage of being simply and quickly calculated, and is often in agreement with other dimensions. For calculation procedure we refer reader to [15]. Some assumptions of change detection techniques are: working with data streams; high dimensional data; limited computation resources. In order to meet this requirements this needs a fast estimation algorithm should be used. The fast implementation of the correlation sum has been proposed by several authors. Wu et al. [20] take advantages of the second frequency moment [1] over the data to calculate the fractal dimension. The algorithm is memory efficient and needs only one pass over the data. It can be used on data streams.

The fractal dimension has inherent properties that are important for change detection. One of the properties of the fractal dimension makes it robust against measurement error but another property imposes limitation on the proposed method. The property from [19] shows that we can use the fractional dimension as a change detector in the presence of noise only if the noise dimension is smaller than the dimension of the signal.

The fractal dimension is immune to affine and bi-Lipschitz transformation [2]. Thus, the change originated by this group of transformation will not be detected by monitoring the dimension.

Time series was split into epochs. An epoch is generated by shifting the estimation window for a fixed number of points. The correlation dimension was estimated on every epoch. For calculation, software from [21] was utilized.

The size of the estimation window influences the dimension value. According to the dimension definition, the window should be as long as possible to get appropriate results. However, we have to remember requirements of the change detection method: the short calculation time and the low resource demand. Enlarging the window will always cause an increase in calculation time. Nevertheless, we could deal with the larger window with constant memory requirements using stream algorithms.

The number of shifting points has the greatest effect on the change detection time. The smaller number of points gives shorter detection time, but increases the number of epochs and simultaneously the calculation time. We must remember that small window size contains small number of new points, so it does not influence the dimension.

In order to make change detection automatic, the EWMA control chart. The change is detected when the plot crosses one of its limit. In the Fig. 3 b) the control charts calculated on the estimated fractal dimension value is depicted.

The proposed method detects change above 5% of the original parameters values, which is an improvement over slight change detection methods described in cited literature. Experiments show that it works well with a presence of noise. However, this method cannot detect all changes so it should be used in conjunction with other change detection methods. Combination of the standard statistical method and the proposed method could enlarge the set of detectable changes.

The parameters needed to be set in this method include the delay, the embedding dimension, the window length and the data shift. The length of the window and new points give smoothness and accuracy. The size of the dataset determines the window. Small datasets do not give the opportunity to pick large values. In case of stream data, the window length is arbitrary. However, this affects computation speed. The smaller window causes larger variance and higher level of false alarms but shortens the change detection time. Thus, the choice of parameters should be a trade off between speed and accuracy.

Recurrence Plot. The recurrence plot is the visual tool representing the recurrence of the system states. Several measures which quantify the structure of the recurrence plot are described in [10]. These and later developed measures [12] have become a popular technique in analyzing experimental data especially in the area of physiology. The big advantage of the RP over the standard time series method is a lack of constraints on the stationarity and data size. These properties makes RPs highly useful in change detection. The big advantage of this method is easy implementation and intuitive understanding.

In order to detect changes in a dynamical system we used the recurrence rate measure. The recurrence rate corresponds to the probability that a specific state

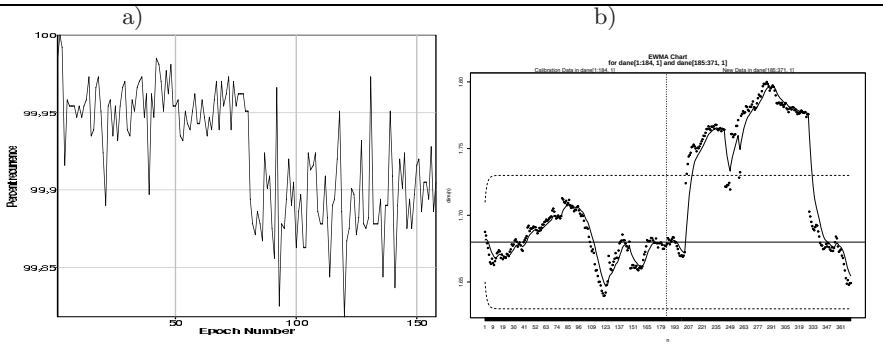


Fig. 3. EWMA control chart for Lorenz system with slight change from 28 to 29.4

will recur. The definition of correlation sum and recurrence rate is almost equal. Other recurrence measures could also be used to accomplished detection task, but they demand more computation.

Our experiments show superior performance of the recurrence rate approach over the fractional dimension. The method can detect smaller changes, even 2% of the original parameter value. Detection is carried out without embedding. Fig. 3 a) shows the recurrence rate plot of the Lorenz system with a slight change. The change is well defined in figure and easily detectable by the control chart.

Fast processing of the data stream can be achieved by taking advantage of the fast estimation method of the correlation dimension. The advantage of recurrence approach originates from the simplicity of the estimation. The method does not contain complex computations which can introduce errors such as multiple averaging and line fitting.

3 Summary

In this paper we proposed a methodology for change detection in dynamical systems with chaos, based on the RBF network, the fractal dimension and the recurrence plot. The proposed method is modular. The method was tested on univariate time series' from the Lorenz system. The unknown slight change was introduced at an unknown time to one of the system parameters. Estimation was performed on small datasets as well as on the data stream. The fractal dimension properties show that the presence of noise which is smaller than the dimension of a signal does not affect the change detection method. All presented approaches provide sufficient information to automate change detection. Using those quantities, it is possible to detect slight changes which are not detectable by standard statistical methods. However, the results show that the recurrence rate approach is superior to the fractal dimension approach and model based approach for the task of change detection. The former method is able to detect smaller changes and does not always require embedding procedures of the time series.

Nevertheless, the proposed approach cannot detect all changes, especially those caused by some groups of transformations and thus, it should not be used as the sole detection method. It should be used in conjunction with other methods in order to detect the larger set of changes.

Acknowledgments. This work was supported by Foundation for Polish Science and Ministry of Science and Higher Education of Poland grant 2006-2009.

References

1. Alon, N., Matias, Y., Szegedy, M.: The space complexity of approximating the frequency moments, pp. 20–29 (1996)
2. Falconer, K.: Fractal Geometry. Wiley, New York (1990)
3. Fraser, A.M., Swinney, H.L.: Independent coordinates for strange attractors from mutual information. Phys. Review A 33 (1986)
4. Gonzalez, F., Dasgupta, D.: Anomaly detection using real-valued negative selection (2003)
5. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall, Englewood Cliffs (1999)
6. Haykin, S., Principe, J.: Making sense of complex world. IEEE Signal Process. Mag. 15, 66–81 (1998)
7. Hively, L.M., Protopopescu, V.A.: Timely detection of dynamical change in scalp eeg signals. Chaos 10 (2000)
8. Ilin, A., Valpola, H.: Nonlinear dynamical factor analysis for state change detection. IEEE Trans. on Neural Network 15(3) (2004)
9. Ruelle, D., Eckmann, J.P., Kamphorst, S.O.: Recurrence plots of dynamical systems. Europhys. Lett. 4 (1987)
10. Webber Jr., C.L., Zbilut, J.P.: Dynamical assessment of physiological systems and states using recurrence plot strategies. J. Appl. Physiol. 78 (1994)
11. Kennel, M.B., Brown, R.: Determining embedding dimension for phase-space reconstruction using a geometrical construction. Phys.Rev. A 45 (1992)
12. Marwan, N.: Encounters With Neighbours - Current Developments Of Concepts Based On Recurrence Plots And Their Applications. PhD thesis (2003)
13. Montgomery, D.C.: Introduction to Statistical Quality Control, 4th edn. Wiley, Chichester (2001)
14. Orr, M.J.: Regularisation in the selection of rbf centres. Neural Computation 7(3) (1995)
15. Ott, E.: Chaos in Dynamical Systems. Cambridge University Press, Cambridge (1993)
16. Principe, J.C., Rathie, A.: Prediction of chaotic time series with neural networks and the issue od dynamic modeling. International Journal of Bifurcation and Chaos (1992)
17. Sauer, T., Yorke, J.A., Casdagli, M.: Embedology. Journal Statistical Physics (1991)
18. Schuster, H.G.: Deterministic Chaos. Weinheim: VGH Verlagsgesellschaft (1988)
19. Tricot, C.: Curves and Fractal Dimension. Springer, Heidelberg (1995)
20. Wong, A., Wu, L.: Fast estimation of fractal dimension and correlation integral on stream data. Inf. Process. Letters 93 (2003)
21. Wu, L., Faloutsos, C., Fracdim, Perl package (January 2001), available at, `http://www.andrew.cmu.edu/~lw2j/downloads.html`

Picard Discretization of Nonlinear Systems: Symbolic or Numeric Implementation?

Jesús Rodríguez-Millán[1], Anna Patete[2], and Carla González[3]

[1] Universidad de Los Andes - Facultad de Ingeniería - Dept. Sistemas de Control
Apartado 11 - La Hechicera, Mérida 5101-A, Venezuela
jrmillan@ula.ve
[2] Tokyo Denki University, Advanced Science and Technology, Ishisaka, Hatoyama, Hiki-gun, Saitama 350-0394, Japan
apatete@furutalab.k.dendai.ac.jp
[3] Universidad Carlos III, Dept. de Sistemas y Automática, Laboratorio de Robótica, Av. De la Universidad 30, Leganés, Madrid 28911, Spain
cguzcate@ing.uc3m.es

Abstract. In this paper we report the numeric implementation of previously proposed symbolic non-standard discretization methods for nonlinear dynamical control systems. We discuss the advantages and disadvantages of both symbolic and numeric implementations, and illustrate them through their applications to case studies.

Keywords: Linear control systems, nonlinear control systems, symbolic discretization, numeric integration, hybrid computation.

1 Discretization of Linear Control Systems

Discretization methods for continuous-time nth-order linear dynamical control systems (LDCS) may be classified in two main groups, namely, *approximated* (ADM) or *exact* (EDM) *discretization methods*, depending on whether one choose to discretize their state-variable differentiable representations:

$$\begin{aligned} &\dot{x} = Ax + Bu,\ x(0) = x_o, \quad x \in \Re^n,\ u(t) \in \Re^k \\ &y = Cx,\ y \in \Re^m, \end{aligned} \tag{1}$$

or their trajectory based integral representations:

$$\begin{aligned} &x(t) = e^{At}x_o + \int_0^t e^{A(t-s)}Bu(s)ds, \quad x \in \Re^n,\ u(t) \in \Re^k, \\ &y(t) = Ce^{At}x_o + \int_0^t Ce^{A(t-s)}Bu(s)ds, \quad y \in \Re^m, \end{aligned} \tag{2}$$

respectively. In ADM first-order derivatives are substituted by first-order differences, what transforms the state-variable differential equation in (1) into the state-variable difference equation (3):

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 121 – 129, 2007.
© Springer-Verlag Berlin Heidelberg 2007

$$x[(k+1)T] = A_d\, x[kT] + B_d\, u[kT],\ \ x[0] = x_o,\ x[kT] \in \Re^n,\ u[kT] \in \Re^k, \qquad (3)$$

where the sampling period $T > 0$ must satisfy Shannon's sampling theorem [1]. Matrices (A_d, B_d) have particular forms:

$$(A_{db}, B_{db}) = ((I - TA)^{-1},\ T(I - TA)^{-1}B), \qquad (A_{df}, B_{df}) = (I - TA,\ -TB), \qquad (4)$$

depending on whether first-order derivatives are substituted by backwards or forwards first-order differences, respectively. Computationally speaking the ADM are the cheapest and easiest to implement, yet they have the worse fitting properties.

Under the assumptions of periodic sampling and zero-order holding, i.e. $u(t) = u(kT),\ kT \le t \le (k+1)T$, the trajectory $x(t)$ in (2) leads to the first-order difference equation

$$x[(k+1)T] = A_{de}\, x[kT] + B_{de}\, u[kT],\quad A_{de} = e^{AT},\ B_{de} = \int_0^T e^{At}dtB, \qquad (5)$$

representing the exact discretization of the continuous-time state-equation in (1). For the same sampling period $T > 0$, EDM has considerable better fitting properties than ADM, especially during the transient-state. However, there are no general constructive symbolic algorithms to compute $e^A,\ A \in M_n,\ n \ge 4$, hence matrices A_{de} and B_{de} in (5). This limitation of the EDM might be partially overcome by approximating matrices A_{de} and B_{de} in (5) by their jth-degree Taylor polynomial approximations:

$$A_{de} \sim A_{dj} = I + TA + \ldots + \frac{T^j A^j}{j!}, \qquad B_{de} \sim B_{dj} = \sum_{i=0}^{j-1} \frac{A^i B}{(i+1)!} T^{i+1}. \qquad (6)$$

The procedure transforming the state-equation in (1) into the difference equation

$$x[(k+1)T] = A_{dj}\, x[kT] + B_{dj}\, u[kT], \qquad (7)$$

will be called the *truncated discretization method* (TDM) of degree jth.

2 Euler Discretization of Nonlinear Control Systems

Most control systems are nonlinear dynamical control systems (NLDCS)

$$\begin{aligned} \dot{x} &= f(x, u),\ x(0) = x_o,\ x \in \Re^n,\ u(t) \in \Re^k, \\ y &= h(x, u), \end{aligned} \qquad (8)$$

which, in general, do not admit to be discretized using neither the backwards ADM, nor the EDM. In fact, the backwards ADM is not extendable to the nonlinear state-equation in (8) because it leads to a non-invertible, with respect to $x[kT]$, first-order nonlinear difference equation. On the other hand, the state-trajectories of most NLDCS (8) cannot be explicitly computed, and therefore they can neither be discretized in exact form.

By the *Euler or standard discretization method* (SDM) for the NLDCS (8) we will mean the natural extension of the forwards ADM to the state-equation in (8):

$$x[(k+1)T] = x[kT] - T\, f(x[kT], u[kT]),\ x[0] = x_o,\ x[kT] \in \Re^n,\ u[kT] \in \Re^k. \quad (9)$$

SDM and/or variations upon it support most numerical integration methods for nonlinear ordinary differential equations [4]. For the SDM to generate discrete-time trajectories with good fitting properties, it is usually necessary to run it at high sampling frequencies. Unfortunately, within the context of the NLDCS, there is nothing like a Shannon's sampling theorem to help us finding the appropriate sampling frequency. The SDM may be metaphorically thought of as "periodic sampling + linear interpolation", and uses to have good steady-state fitting properties.

3 Picard Iterations

In spite of the fact that the trajectories of most NLDCS (8) are not explicitly and symbolically computed, we may think of them as differentiable curves x: $\Re \to \Re^n$ such that:

$$x(t) = x_o + \int_0^t f(x(s),u(s))ds. \quad (10)$$

When the vector field $f(x, u)$ in the NLDCS (8) is Lipschitz continuous [4], Picard iteration method provides a constructive, though infinite, algorithm:

$$\begin{aligned} \phi_o(t) &= x_o \\ \phi_k(t) &= x_o + \int_0^t f(\phi_{k-1}(s),u(s))ds, \quad k = 1, 2, \ldots \end{aligned} \quad (11)$$

to compute a trajectory ϕ: $\Re \to \Re^n$, namely $\phi = lim\ \phi_k$, $k \to \infty$ [4], satisfying condition (10). For small enough $t > 0$, Picard iteration method provides local approximate trajectories with remarkable good fitting properties, after just a few iterations, what led us to propose the Euler-Picard discretization method (EPDM) in [5]. EPDM may be summarize as "periodic sampling + Picard interpolation". That the Picard iteration method actually generate meaningful results may be appreciate in the following case study.

Case Study 1. Picard approximate trajectories for the LDCS (1) around x_0

Let us suppose that $f(x, u)$ in (11) is given by $f(x, u) = Ax + Bu$, and $x(0) = x_o$. Then, the straightforward calculation of Picard iterations $x_k(t)$ in (11) generates the following results:

$$\begin{aligned} x_o(t) &= x_o \\ x_1(t) &= (I + tA)x_o + B\int_0^t u(s)ds \end{aligned} \quad (12\text{-a})$$

$$x_2(t) = (I + tA + \frac{t^2 A^2}{2})x_o + B\int_0^t u(s)ds + AB\int_0^t \int_0^v u(s)dsdv$$

……

$$x_k(t) = (I + tA + \frac{t^2 A^2}{2} + \ldots + \frac{t^k A^k}{k!})x_o + B\int_0^t u(s)ds + AB\int_0^t \int_0^v u(s)dsdv \qquad (12\text{-b})$$

$$+ \ldots + A^{k-1}B\int_0^t \int_0^{v_{k-1}} \ldots \int_0^{v_1} u(s)ds\, dv_1 \ldots dv_{k-1}.$$

Let us know sample the control signal $u(t)$ at $t = 0$, and zero-hold it during the sampling-period $[0, T]$. Then,

$$x_k(T) = (I + TA + \frac{(TA)^2}{2} + \ldots + \frac{(TA)^k}{k!})x_o + (TB + \frac{T^2}{2}AB + \ldots + \frac{T^k}{k!}A^{k-1}B)u(0), \quad (13)$$

or, equivalently:

$$x_k[T] = A_{dk}\, x[0] + B_{dk}\, u[0], \qquad (14)$$

where A_{dk} *and* B_{dk} are the very same matrices we would obtain if one applied the kth-order TDM to the initial-value problem $\dot{x} = Ax + Bu,\ x(0) = x_o$, during the first sampling-period [0, T].!

4 Symbolic Integral Methods Are Computationally Expensive

As it transpires from (11) and Case Study 1, one might expect the Picard iteration method to result in a discretization method with very good fitting properties, yet computationally expensive. In fact, on the one hand, the structural similarities between (10), the actual solution of the state-equation in (8), and the kth Picard approximate trajectory in (11) speaks in favour of the good fitting properties of Picard trajectories in small enough neighborhoods of the initial state x_o. On the other hand, the very same structure of Picard iterations warns us against the difficulties we might find on computing integrals of nested functions as the order of Picard iterations increases. Simple challenging examples abound.

Case Study 2. The one-dimensional simple pendulum without friction
Let us consider a one-dimensional simple pendulum $\dot{x} = Sin[x(t)],\ x(0) = x_o$. Even though the first three Picard iterations are fairly simple to computed:

$$x_o(t) = x_o,$$

$$x_1(t) = x_o + \int_0^t Sin[x_o]ds = x_o + t\, Sin[x_o], \qquad (15\text{-a})$$

$$x_2(t) = x_o + \int_0^t Sin[x_o + s\ Sin[x_o]]ds$$
$$= x_o + Cot[x_o] - Csc[x_o]\ Cos[x_o + t\ Sin[x_o]], \tag{15-b}$$

it is not possible to compute the fourth Picard iteration, involving integrals like

$$\int_0^t Sin[a\ Sin[b + c\ s]]ds, \text{ and } \int_0^t Cos[a\ Cos[b + c\ s]]ds, \tag{16}$$

in terms of standard mathematical functions [8]. Just as a matter of reference, *Mathematica 5.0* needs 0.000033, 0.004024, and 0.030924 seconds of CPU time to compute the first three Picard iterations above, on a 1.33 GHz PowerPC G4 Macintosh computer.!

Case Study 3 below shows that the difficulties to compute third or higher order Picard iterations are by no means exclusive of trigonometric systems.

Case Study 3. A one-dimensional tank

Consider the first-order NLDCS $\dot{x} = Sqrt[x(t)],\ x(0) = x_o$, which may be thought of as a normalized model of a single tank. Again, the first three Picard iterations around the initial state may be worked out in terms of standard mathematical functions:

$$x_o(t) = x_o,$$
$$x_1(t) = x_o + \int_0^t Sqrt[x_o]ds = x_o + t\ Sqrt[x_o],$$
$$x_2(t) = x_o + \int_0^t Sqrt[x_o + t\ Sqrt[x_o]]ds \tag{17}$$
$$= \frac{1}{3}\left(x_o + 2\left(t\ Sqrt[t\ Sqrt[x_o] + x_o] + Sqrt[t\ x_o^{3/2} + x_o^2]\right)\right).$$

but $x_3(t)$ does not. The CPU time needed to compute the first three Picard iterations in (17) were 0.000029, 0.126742, and 0.436168 seconds, respectively.!

Case Studies 2 and 3 clearly state that the main obstacle to apply the EPDM to arbitrary families of NLDCS is the recursive symbolic computation of integrals of nested nonlinear functions. This notwithstanding, it is also worth to remark that in the case of polynomial NLDCS, the recursive symbolic computation of integrals of nested polynomials may be time consuming, but is also simpler and, in principle, computable in finite time. This structural characteristic of polynomial systems is precisely what led us to propose the Euler-Taylor-Picard discretization method (ETPDM) in [5], where the main idea is to approximate non-polynomial NLDCS (8) by polynomial ones, and then discretize these last ones according to the EPDM.

5 Always Compute as Few Picard Iterations as Possible

In previous works [6, 7] we reported several improvement on the basic EPDM and ETPDM described above, all of which were mainly oriented to reduce computation times, and finding an appropriate sampling-period for discretization purposes, while keeping all computations symbolic. Two important general conclusions came out from this search: (i) whatever be the prescribed fitting criterion for the discretized trajectories, keep the order of Picard iterations as low as possible, and (ii) even low-order Picard iterations may not be symbolically computable for NLDCS with seemingly simple structures.

That the time needed to compute Picard iterations $x_n(t)$ might explode as their order n grows higher is shown in Table 1, for the simplest most one-dimensional polynomial NLDCS $\dot{x} = x^n$, $x(0) = x_o$, $n = 0, 1, 2, \ldots, 10$.

Table 1. CPU time to compute first five Picard Iterations for the NLDCS $\dot{x} = x^n$

n	$x_1(t)$	$x_2(t)$	$x_3(t)$	$x_4(t)$	$x_5(t)$	r
1	0.000022	0.068579	0.217699	0.456264	0.780355	3.5846
2	0.000029	0.069939	0.307697	0.901289	2.751750	8.9431
3	0.000029	0.068370	0.400586	0.534241	6.769180	16.898
4	0.000031	0.069049	0.313771	0.500199	50.14360	159.810
5	0.000031	0.077465	0.359631	0.627330	?	?
6	0.000026	0.066514	0.386109	0.587365	?	?
7	0.000028	0.080966	0.777241	8.436650	?	?
8	0.000029	0.081266	0.959847	13.18400	?	?
9	0.000031	0.079248	1.147660	18.23370	?	?
10	0.000028	0.081776	0.802239	34.84030	?	?

From the practical point of view the first significant Picard iteration to compute is the third one, because the first two are always a constant and a first-order polynomial, respectively. The last column of Table 1 shows the quotient r of the CPU times needed to compute the fifth and the third Picard iterations, for $1 \leq n \leq 4$. The fifth Picard iteration is not symbolically computable in reasonable times for $n \geq 5$. According to the index r, it would take nearly 160 times longer to compute the fifth than the third Picard iteration when $n = 6$. So, in order to satisfy a prescribed fitting criterion, it will take shorter to increase the sampling-frequency and use a third or fourth-order EPDM, than to keep the sampling-frequency lower and use a fifth-order EPDM.

6 Numeric Implementation of Picard Iteration Method

And it may even be yet, that depending on the nonlinearities involved, neither third Picard iterations were symbolically computable in terms of standard mathematical functions, or that it took too long to compute the integrals for the results to be of

practical use. In fact, along the development of our work we have been collecting models of physical systems that could not be discretized using the EPDM, the ETPDM, or any of their already reported improved versions: adaptive Picard degree discretization, adaptive Taylor degree discretization, adaptive mono-frequency discretization, adaptive multi-frequency discretization, etc.

Transforming symbolic discretization methods into numerical discretization method is a conceptually straightforward operation, because it is all about computing the definite integrals involved either symbolically or numerically. Moreover, given that all our work have been developed using *Mathematica*® as a computational framework, in a lot of cases it is only needed to switch the symbolic integration functions by the numerical ones, to convert previous symbolic developments into numerical ones.

That switching symbolic integration by numerical integration actually expand the set of NLDCS we can discretize using the EPDM, the ETPDM, and their variations is shown in the following Case Studies, which has acted as benchmark for our work.

Case Study 4. A Cascade of Two Tanks

Let us consider a model of a cascade of two tanks [9], where x_1 and x_2 are the levels of the first and the second tank, respectively, and $u(t)$ is the input flow to the uphill tank:

$$\begin{aligned} \dot{x}_1 &= -\frac{K_1}{C_1} Sqrt[x_1] + \frac{1}{C_1} u(t) \\ \dot{x}_2 &= \frac{K_1}{C_2} Sqrt[x_1] + \frac{K_2}{C_1} Sqrt[x_2]. \end{aligned} \tag{18}$$

Tables 2 and 3 summarize the CPU times needed to compute the discretization of the open-loop dynamics of the NLDCS (18), when $u(t) = K_2X$, so that $x_2(\infty) = X_2$, using the EPDM and the ETPDM, respectively. In both tables n represents the nth Picard iteration, whereas b represents the degree of the Taylor polynomial approximation involved in the ETPDM. As it is clearly establish in these tables, the numeric implementations of the EPDM and the ETPDM did succeed on computing the third and fourth Picard iterations, whereas the symbolic implementations failed. As it is to be expected, the higher the order of the Picard iteration, the better the fitting properties of the corresponding discretization. Yet, there is not enough space in this paper to include all the corresponding simulations.

Table 2. EPDM

n	Symbolic	Numeric
2	0.992	0.19
3	Aborted	1.542
4	Aborted	8.713

Table 3. ETPDM

n	b	Symbolic	Numeric
2	3	0.691	0.27
3	3	Aborted	2.113
4	3	Aborted	11.827

Tables 4 and 5 summarize the CPU times needed to compute the discretization of the closed-loop system consisting of the NLDCS (18) and a nonlinear extended state-feedback controller. Consistently with the previous results for the open-loop dynamics, the numeric implementations of the EPDM and the ETPDM succeeded where the symbolic ones failed.

Table 4. EPDM

n	Symbolic	Numeric
2	2.564	0.530
3	Aborted	4.145
4	Aborted	22.473
5	Aborted	97.620

Table 5. ETPDM

n	b	Symbolic	Numeric
2	3	2.444	0.691
3	3	Aborted	4.076
5	3	Aborted	100.064

7 Conclusions and Further Work

The results reported in this work clearly show that the numerical implementation of the EPDM and ETPDM allows extending the applicability of the proposed discretization methods to bigger classes of NLDCS and, in some cases, to reduce the CPU time needed to work calculations out. Similar results for other families of NLDCS are reported in [10, 11]. Yet, numeric is not a panacea, and raises multiple questions, demanding further attention and research: convergence of algorithms, control over the interpolations involved, invertibility problems, etc.

References

[1] Oppenheim, A., Willsky, A., Hamid, S.: Signals and Systems, 2nd edn. Prentice-Hall, Englewood-Cliffs (1997)
[2] Isermann, R.: Digital Control Systems, Fundamentals, Deterministic Control, vol. 1. Springer, Berlin (1989)
[3] Rodríguez-Millán, J.: Sistemas Dinámicos (E-book), Universidad de Los Andes, Mérida (to appear, 2007)
[4] Coddington, E., Levinson, N.: Theory of Ordinary Differential Equations. Tata McGraw-Hill Publishing Co, New Delhi (1977)
[5] Rodríguez-Millán, J., González, C.: Three Mathematica Supported Proposals for the Discretization of Nonlinear Control Systems, Nonlinear Analysis 63, e617-628 (2005), Available online at, http://www.sciencedirect.com
[6] Rodríguez-Millán, J., González, C., Patete, A.: Analysis, Synthesis and Mimetization of Nonlinear Dynamical Control Systems Using Mathematica. In: Quesada-Arencibia, A., Moreno-Díaz, j.R., Rodríguez, J. (eds.) Cast and Tools for Robotics, Vehicular and Communication Systems, Eurocast 2005 - Extended Abstracts. IUCTC Universidad de Las Palmas de Gran Canarias, Las Palmas, pp. 316–320 (2005)
[7] Rodríguez-Millán, J., González, C., Patete, A.: Improved Non-standard Discretization Methods for Nonlinear Dynamical Control Systems. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, Springer, Heidelberg (2005)

[8] Wolfram, S.: The Mathematica Book, 5th edn. Wolfram Media, USA (2003)
[9] Patete, A.: Discretization of Nonlinear Systems, Case Study: A Cascade of Two Tanks, Universidad de Los Andes, Postgrado de Control y Automatización, Mérida (2007)
[10] Patete, A.: Discretization of Nonlinear Systems, Case Study: Two Pendulum Mounted on a Chariot, Universidad de Los Andes, Postgrado de Control y Automatización, Mérida (2007)
[11] Patete, A., Discretization of Nonlinear Systems, Case Study: The Centrifugal Pendulum, Universidad de Los Andes, Postgrado de Control y Automatización, Mérida (2007)

An Object-Oriented and Generic Compiler Generator

Michael Pitzer and Heinz Dobler

University of Applied Sciences, Hauptstr. 117, 4232 Hagenberg, Austria
{michael.pitzer, heinz.dobler}@fh-hagenberg.at

Abstract: Object-oriented software development has become the de-facto standard programming paradigm used in modern software systems. Additionally genericity has grown more popular since the enhancement of Java and C#. This paper attempts to reconsider the principles of compiler construction from this modern, object-oriented point of view. We present a multi-paradigm, mainly object-oriented and generic approach for creating a compiler generator using a combination of the Interpreter pattern and the Visitor pattern. A prototype of such an object-oriented and generic compiler generator has also been developed using C# 2.0 and will serve as a reference to explain the design throughout this paper.

Keywords: Compiler, Compiler Generator, Design Patterns, Interpreter, Visitor, EBNF, Attributed Grammars, Genericity.

1 Introduction and Motivation

Programming using different programming paradigms and languages as well as basics of formal language theory and compiler construction form an essential part in computer science and especially in software engineering education. This paper is the result of work mainly carried out in the context of a bachelor thesis, which had the aim to investigate the applicability of the object-oriented and generic programming paradigms in the field of compilers and compiler generators. The fist step was to study existing techniques for both, compilers and compiler generators. Aho and Ullman in [1] provide a sound overview over the principles of compiler construction. Then, existing compiler generators, traditional ones like lex and yacc [5] (aka flex and bison from GNU) as well as modern ones like JavaCC [9] and Coco-2 [7] were examined. All mentioned generator examples turned out to follow mainly the imperative/procedural programming paradigm. The investigations turned up that all these tools have one aspect in common: they generate old-fashioned procedural code. In this paper, the authors will present a new approach to port this old idea to modern programming paradigms, specifically the object-oriented and generic paradigms.

2 Object-Oriented and Generic Compiler Generator

The main idea of this approach is using object-oriented programming and generic classes in a compiler generator as well as in the compilers generated by it. This way, the readability of the code shall be improved and it should be easier to maintain.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 130 – 137, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Provided a proper input grammar, the compiler generator could be bootstrapped, that means, it would be able to generate itself. In [4] is stated that "two of the most useful abstractions used in modern compilers are context-free grammars, for parsing, and regular expressions, for lexical analysis." Having this in mind, the hybrid compiler generator will also make use of both techniques. Like in existing compiler compilers, (E)BNF (see section 3.1) is used to specify the grammar for the compiler to be generated. The first occasion where the difference to procedural compilers will show up, is the syntax tree, which will be created from an input stream. In this case, we are literally talking about a tree: a tree of objects representing specific parts of the source grammar. By combining the design patterns Interpreter and Visitor [2] we will be able to perform operations on this tree. So, to understand the design of the presented compiler generator, it will be necessary to learn about EBNF, the Interpreter and Visitor patterns as well as generic programming in the following sections.

2.1 (E)BNF Notation

EBNF is an extension by Niklaus Wirth to BNF, short for *Backus-Naur-Form*. It is a standard notation for describing context-free grammars. That is why many compiler generators, just like the one presented in this paper, use variations of this notation to define the grammar of the source language for a compiler to be generated. This assures that the compiler generator is able to deal with any given context-free grammar. To define rules for a grammar, EBNF uses the syntactic elements alternative |, option […], repetition { … }, and sequence, which is represented just by a blank. Brackets are also used to declare precedence. Let us consider an example in the XML realm:

```
XMLNode = '<' ident '>' XMLElement '</' ident '>'.
XMLElement = { text | XMLNode }.
```

The above example shows a simplified grammar for parts of an XML document in EBNF. The input for the presented hybrid compiler generator will be exactly such a grammar. The main idea is the mapping of all syntactic elements of EBNF as well as all terminal and non-terminal symbols of a grammar to specific classes. Objects of these classes will be able to parse instances of the respective parts of the grammar. For example, objects of the class *Rule* representing a non-terminal symbol (left-hand side) with its replacement (right-hand side of the grammar rule) will be able to parse parts of a sentence conforming to this rule of the grammar and will additionally represent this rule in the form of an object tree.

2.2 Design Pattern Interpreter

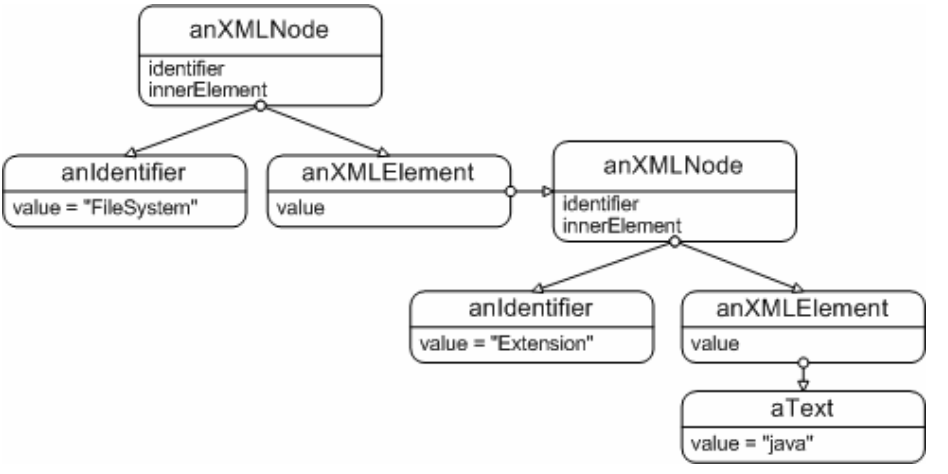
The Interpreter pattern is an easy way to represent sentences of any language in an object-oriented programming language. As described in [2], the Interpreter pattern "uses a class to represent each grammar rule. Symbols on the right-hand side of the rule are instance variables of these classes." That is how non-terminal symbols are represented. Now, we consider each terminal symbol of the grammar, a rule of the form

```
SymbolName = 'symbol'.
```

We can now treat terminal symbols the same way as non-terminal symbols. This rule just gives a name to the class that represents the symbol. Now, that it is

possible to represent all terminal and non-terminal symbols of a grammar as objects, we are able to represent any sentence of the source language by assembling a syntax tree with concrete objects of these classes, like it is done for XML nodes in Illustration 1.

Illustration 1. XML syntax tree using the above grammar



Each element of the syntax tree is derived from a common abstract root class, which defines an *Interpret* method. All derived classes implement this method, in which the desired transformations are performed and the output of the compiler, for example machine code, is created. By traversing the syntax tree and calling the *Interpret* method for each object in the tree, we can perform these operations on the whole input.

Illustration 2. Class diagram for an XML interpreter according to the previous examples

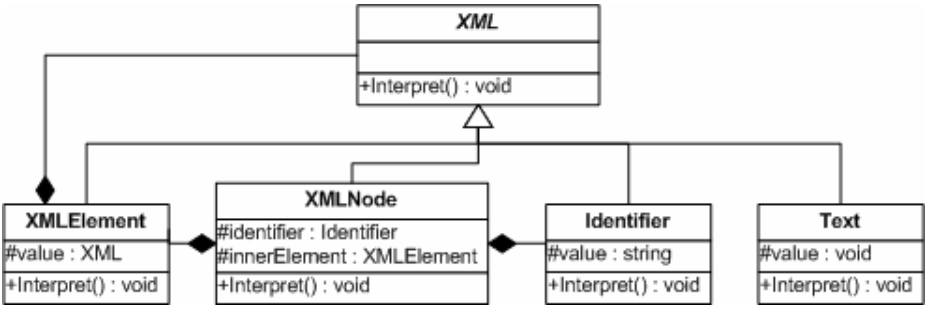


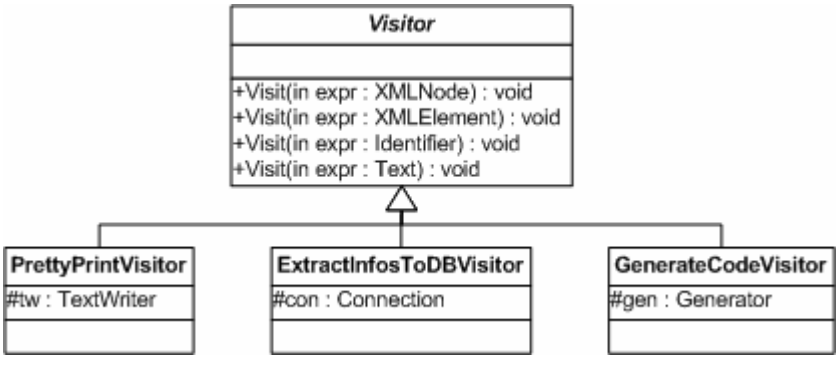
Illustration 2 shows how all symbols of the grammar are represented by classes derived from the abstract root class *XML*, which override the *Interpret* method. The two classes on the side represent non-terminal symbols. The two classes on the right represent terminal symbols, or rather terminal classes. The Composite pattern [2] is used to map recursive rules in the grammar to the class structure.

Often, the user of such a compiler will want to perform various operations on such a tree. For example he might want to generate machine code, pretty-print the code or measure the complexity of the code represented by the tree. He would have to define an additional *Interpret* method in the root class and implement it in all derived classes. The more classes there are, the more time it will take to do that. It gets rather impractical for complex grammars. Gamma et al. [2] note that "the Interpreter pattern works best when the grammar is simple. For complex grammars, the class hierarchy for the grammar becomes large and unmanageable." By combining the Interpreter with the Visitor pattern, however, we can workaround this problem.

2.3 Design Pattern Visitor

According to [2], with the Visitor pattern it is possible to define operations that should be performed on the elements of an object structure without changing the classes of these elements. This means, we can add new operations to the syntax tree without having to mess with the classes of the objects that make up the syntax tree. Each operation is hereby defined as a single Visitor class.

Illustration 3. A Visitor class hierarchy that executes various operations on an XML tree



All *Visitor* classes that will be working on the same grammar are derived from a common root class, an abstract *Visitor* class. For each symbol of the grammar, a *Visit* method is declared in the abstract *Visitor* class and implemented in all derived concrete *Visitor* classes. To perform an operation on a syntax tree, the tree is traversed, and a *Visitor* object visits each object of the tree. Therefore, the *Interpret* method from the Interpreter pattern is replaced by an *Accept* method. When visiting an element of the tree, the *Visitor* object calls the element's *Accept* method. The visited object just calls back the appropriate *Visit* method of the *Visitor* object, where the operation is implemented, and passes itself as parameter. By receiving this parameter, the Visitor now possesses all required information to perform its operation. So, we have now replaced all the *Interpret* methods with one single *Accept* method that accepts different classes of Visitors to perform different types of operations.

Gamma et al. in [2] suggest unambiguous names for each *Visit* method, like *VisitXMLNode*, *VisitText*, or *VisitIdentifier*, but we can also just overload the *Visit*

method as shown in Illustration 3. That way, we will be able to implement *Visit* methods for the abstract elements of our grammar. For example, a *Visit* method that takes a terminal symbol could implement the default behaviour of an operation for all terminal symbols. We would not have to implement all operations for all elements of a grammar. Dynamic binding provides that the correct *Visit* method will be called at run-time.

Now, if we want to add new operations to our syntax tree, we only have to derive a new class from the abstract *Visitor* class and implement the required *Visit* methods for this operation. However, for the ease of adding new operations, we sacrifice the ease of extending the object structure. If new rules were added to our grammar, we would have to add a new *Visit* method to each of the *Visitor* classes. However, this is rarely the case. From [2] we learn that "the classes defining the object structure rarely change, but you often want to define new operations over the structure."

2.4 Generic Aspects of the Compiler Generator

To represent the syntactic elements of EBNF, ordinary classes will not suffice. Alternative, sequence, option, and repetition must be implemented as generic classes, because we do not know in advance which symbols will be used within these elements. For example, if alternative was implemented with conventional classes, we would have to implement new *Alternative* classes for almost each occurance of | in the grammar of the source language. Each of these classes would be limited to two specific types. Generic classes, however, can be instantiated with any concrete type and allow us to map each alternative of a grammar to an instantiation of the generic *Alternative* class. Furthermore, if a rule consists of more than two alternatives, interlocking instantiations of the generic class can still represent this rule. The following classes can be used to put together the right-hand side of any rule given in EBNF:

- `Sequence<T1, T2>` to represent … T1 ○ T2 …,
- `Alternative<T1, T2>` to represent … T1 | T2 …,
- `Option<T>` to represent … [T] … and
- `Repetition<T>` to represent … { T } …. .

The type parameters *T1* and *T2* respectively *T* will then be replaced by concrete types. For example the rule

```
A {B | C}.
```

will be mapped to the class

```
Sequence<A, Repetition< Alternative<B, C> > >
```

3 Design of an Object-Oriented and Generic Compiler Generator

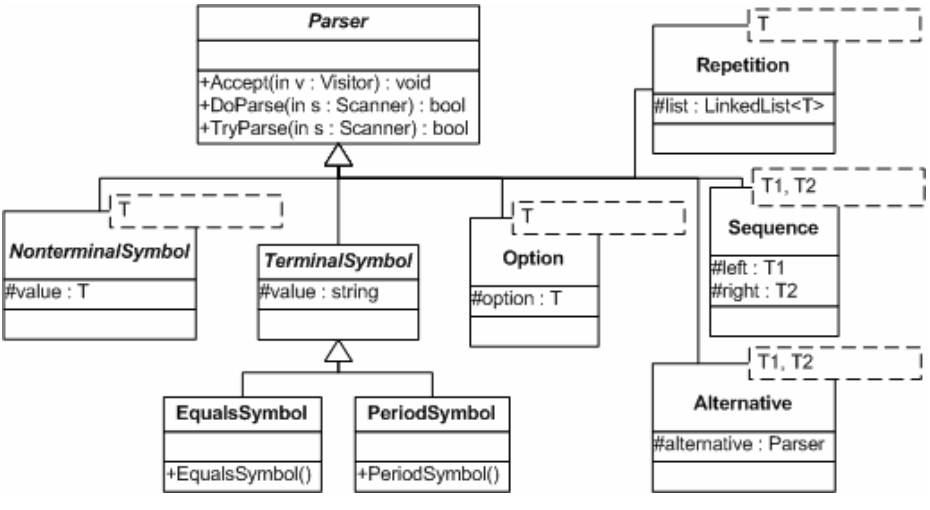
As has been mentioned in section 3.1, EBNF is used in compiler generators to describe any context-free language. So, our object-oriented and generic compiler

generator has to implement the Interpreter pattern for EBNF, similar to ideas formulated in [3]. Since the input for the compiler generator will be grammars denoted in EBNF, we can describe the input for our compiler generator by expressing EBNF in EBNF.

```
Grammar = Rule { Rule }.
Rule = nonterminalIdent '=' Expr '.'.
Expr = Term { '|' Term }.
Term = Fact { ' ' Fact }.
Fact = '•' | terminalIdent | nonterminalIdent |
       '[' Expr ']' |
       '{' Expr '}' |
       '(' Expr ')'.
```

For each syntactic element of EBNF, namely alternative, sequence, option, and repetition, a generic class is derived from the abstract root class *Parser* in our object-oriented compiler generator. Also, the two classes *TerminalSymbol* and *NonterminalSymbol* are derived from *Parser*. These classes form the core of our object-oriented compiler generator.

Illustration 4. Class diagram of a parser for EBNF



When parsing a grammar in EBNF, all the generator does, is to add a new class derived from *TerminalSymbol* for each terminal symbol found in the input grammar. In the case of EBNF that is just a period symbol and an equals symbol like shown in Illustration 4. Accordingly, for each rule of the input grammar, a new class is derived from *NonterminalSymbol*, instantiated with a combination of instantiations of *Alternative*, *Sequence*, *Option*, and *Repetition* that resembles the right-hand side of the rule. The following listing shows the class declarations that the compiler generator creates for the first three rules of the above grammar of EBNF.

```
class Grammar : NonterminalSymbol <Sequence<Rule,
                                Repetition<Rule>>>

class Rule : NonterminalSymbol <Sequence<Ident,
             EqualsSymbol, Expr, PeriodSymbol>>

class Expr : NonterminalSymbol <Sequence<Term,
           Repetition<Sequence<BarSy, Term>>>>
```

So for the generator it is rather easy to generate the code required for parsing a sentence of the specified language. All necessary information to perform this task can be found in the EBNF grammar of the source language. However, this is just the front-end of the compiler, how can we describe the transformations and operations that we want to apply on the syntax tree? The EBNF notation will have to be extended with attributes to add semantic information to the grammars as described in [8]. This is usually done by writing semantic actions next to the element of the grammar that will trigger the action when it gets recognized in the input stream.

As discussed above, in our object-oriented approach we use *Visitor* classes to define operations on the syntax tree. The compiler generator will generate as many *Visitor* classes as operations are defined in the input. As a result, semantic actions have to be qualified via the name of the Visitor that they belong to. This way, the compiler generator will be able to add the actions to the appropriate *Visitor* class. The following example shows what the notation looks like for addition and subtraction in arithmetic expressions. Two *Visitor* classes have to be generated: a *CalculateVisitor* that calculates the result of the arithmetic expression, and a *PrintVisitor* that simply prints out the expression.

```
Expr = Term    CalculateVisitor:  left = stack.pop();
       { ('+'  CalculateVisitor:  operation = Plus;
               PrintVisitor:      writer.write(" + ");
       | '-'   CalculateVisitor:  operation = Sub;
               PrintVisitor:      writer.write(" - ");
       ) Term  CalculateVisitor:  right = stack.pop();
                                  if (operation == Plus)
                                       left += right;
                                  else if (operation == Sub)
                                       left -= right;
       }       CalculateVisitor:  stack.push(left);
       .
```

The position of a semantic action defines at which moment in the parsing process the action will be executed: An action will be executed, when the symbol after which it is located is successfully parsed in the input stream. So, always when a plus symbol is found in the input stream, the *PrintVisitor* will execute the action that prints " + ". This notation, however, is just a suggestion, there is no “official standard” for extending EBNF with semantic actions.

4 Implementation of a Prototype

As a proof of concept the authors have developed a prototype of such an object-oriented and generic compiler generator in the course of a bachelor thesis [10]. It has been implemented using Microsoft .NET and C# 2.0 [6]. The class structure presented in the previous chapter corresponds to the implementation of the prototype. However, two versions of the back-end are available. One generates C# source code like conventional compiler generators. A second version is available that directly generates and loads Common Intermediate Language code using *CodeDom*. C# has been chosen over Java because Java's type erasure during compilation limits the use of generic classes in cases where you have to rely on reflection to create instances of these classes.

5 Summary

This paper described an attempt to apply modern programming paradigms, especially the object-oriented and generic paradigm to the field of compiler construction. The design of a compiler generator based on these paradigms has been shown. The presented design uses a combination of the Interpreter and Visitor pattern to represent sentences of a language and operations working on these sentences. Generic classes are used to represent the syntactic elements of EBNF, alternative, sequence, option, and repetition, and to glue together classes that correspond to the rules of a given grammar. A variation of attributed grammars that allows the definition of various operations on a grammar has been presented. Finally, a prototype of such an object-oriented and generic compiler generator has been developed to show the applicability of the presented ideas.

References

1. Aho, A.V., Ullman, J.D.: Principles of Compiler Design. Addison-Wesley, London, UK (1977)
2. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns–Elements of Reusable Object-Oriented Software. Professional Computing (1995)
3. Lorenz, D.H.: Tiling Design Patterns–A Case Study Using the Interpreter Pattern. ACM SIGPLAN Notices 32, 206–217 (1997)
4. Appel, A.W., Palsberg, J.: Modern Compiler Implementation in Java. Camebridge University Press, Camebridge (2002)
5. Levine, J.R., Mason, T., Brown, D.: Lex & Yacc. O'Reilly & Associates (1992)
6. Thai, T.L., Lam, H.: .NET Framework Essentials. O'Reilly (2002)
7. Dobler, H.: Coco-2: A new compiler compiler. SIGPLAN Notices 25 (1990)
8. Knuth, D.E.: Semantics of context-free languages. Springer, New York (1967)
9. McFlynn, D., Weissman, P.F.: Using JavaCC and SableCC. 4UPress (2004)
10. Pitzer, M.: An object-oriented compiler generator. Bachelor Thesis (2006)

A k-NN Based Perception Scheme for Reinforcement Learning*

José Antonio Martín H.[1] and Javier de Lope[2]

[1] Dep. Sistemas Informáticos y Computación
Universidad Complutense de Madrid
jamartinh@fdi.ucm.es
[2] Dept. of Applied Intelligent Systems
Universidad Politécnica de Madrid
javier.delope@upm.es

Abstract. A perception scheme for Reinforcement Learning (RL) is developed as a function approximator. The main motivation for the development of this scheme is the need for generalization when the problem to be solved has continuous state variables. We propose a solution to the generalization problem in RL algorithms using a k-nearest-neighbor pattern classification (k-NN). By means of the k-NN technique we investigate the effect of collective decision making as a mechanism of perception and action-selection and a sort of back-propagation of its proportional influence in the action-selection process as the factor that moderate the learning of each decision making unit. A very well known problem is presented as a case study to illustrate the results of this k-NN based perception scheme.

Keywords: Reinforcement Learning, k-Nearest-Neighbors, Collective Decision Making.

1 Introduction

Reinforcement Learning (RL) [1,2,3,4] is a paradigm of modern Machine Learning (ML) which uses rewards and punishments to guide the learning process. One of the central ideas of RL is learning by "direct-online" interaction with the environment. In this sense this is the key difference from supervised machine learning in which the learner is told what actions to take. Instead of that, in RL the agent (learner) acts autonomously and only receives a scalar reward signal that is used for evaluate what so good is the actual behavioral policy. The framework of RL is designed to guide the learner in maximizing the average reward in the long run. One of the consequences of this learning paradigm is that the agent must explore new behavioral policies because there is no supervisor that tell what actions to do, thus, the trade off between *exploration* and *exploitation* is a key characteristic of RL. Typically, exploration procedures select actions

* This work has been partially funded by the Spanish Ministry of Science and Technology, project DPI2006-15346-C03-02.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 138–145, 2007.
© Springer-Verlag Berlin Heidelberg 2007

following a random distribution in order to gain more knowledge of the environment reward function while exploitation selects the bests possible actions (using the actual knowledge of the environment reward function) in order to maximize the rewards in the long run.

One of the main drawbacks of classical reinforcement learning is the combinatorial explosion when multiple states variables and multiple actuators are needed to optimally control a complex agent in a complex dynamical environment. This situation produces a huge table of $states \times actions$ pairs which, in the best case, assuming enough memory to store the table, the learning algorithm needs a huge amount of time to converge, indeed the problem could become intractable. Thus the main direction for avoiding this is generalization.

The k-nearest neighbor (k-NN) is a method for classifying objects based on closest training examples in the feature space [5,6]. The training examples are mapped into a multidimensional feature space. The space is partitioned into regions by class labels of the training samples. A point in the space is assigned to a class if it is the most frequent class label among the k nearest samples used in the training phase. The determinant parameters in k-NN techniques are the number k which determines how many units are to be considered as the neighbors, and the distance function, generally the Euclidian distance is used.

In this work we present a solution to the generalization problem in RL algorithms using a k-nearest-neighbor pattern classification (k-NN). By the means of the k-NN technique we investigate the effect of multiple decision makers as the main mechanism of perception and action-selection and a sort of back-propagation of its proportional influence in the action-selection process as the factor that moderate the learning of each decision making unit. The effect of collective decision making in the action selection mechanism forces generalization but some interesting characteristics of collective decision making such as the relevance and quality of each decision making unit must be analyzed in order to obtain a good learning algorithm.

2 Description of the Proposed Method

It has been proved that the Q-Learning algorithm [7] converges with probability 1 to the optimal policy (behavior) under the assumption of a discrete state space with a tabular representation, also see [4, p.148]. However, the tabular approach suffers the curse of dimensionality and indeed it is not possible its application for continuous states spaces, indeed, there are no proofs of convergence when the state space is not discrete despite it has been proved at least empirically that nearest neighbor converges to a near optimal solution (when it can be applied!). One of the most active research areas in present Reinforcement Learning is how to deal with continuous problems. Thus, we are proposing and studying the use of a k-NN based perception scheme to handle continuous state spaces (Fig. 1 and 2). Fig. 2 depicts the perception scheme at a given instant in the learning process for a value of k=5 where columns $(a1, a2)$ represent the expected future reward of each action for each classifier $c_i \in [1, ..., 5]$, column (d) is the Euclidian

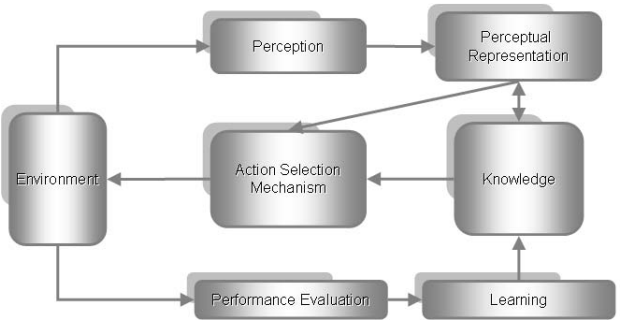
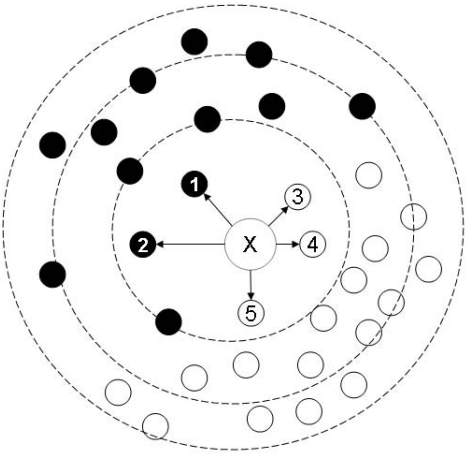


Fig. 1. Architecture of a Reinforcement Learning Agent with perceptual representation in order to handle continuous state spaces



	a1	a2	d	w	a1 total	a2 total
1	1,00	-1,00	1,75	0,25	0,25	-0,25
2	2,00	-2,00	2,50	0,14	0,28	-0,28
3	0,50	0,75	1,00	0,50	0,25	0,38
4	-2,00	2,75	1,00	0,50	-1,00	1,38
5	-1,00	1,13	1,50	0,31	-0,31	0,35
					-0,54	1,58

Fig. 2. k-NN based perception scheme at a given instant in the learning process for a value of k=5

distance from the current mapped perception point (X) to each classifier (c_i) and the values of column (w) represents the weights of each classifier based on the euclidian distance following the formula:

$$w_i = \frac{1}{1 + {d_i}^2} \tag{1}$$

Thus, less distance means more weight, that is, more influence in the action selection process. Then, the action selection mechanism can be realized by weighting each reward prediction and to determine based on this weighted reward prediction which action is the best choice in order to maximize the future reward as shown in (2).

$$\text{best action} = MAX\left[\sum w_i \times a_{i,1}, ..., \sum w_i \times a_{i,n}\right] \tag{2}$$

Also, based on this procedure we define a learning rule in which all classifiers involved in each action selection tournament are modified according to its influence in the action selection mechanism. Thus, given the classical learning rule (Q-learning rule):

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_a Q(s_{t+1}, *) - Q(s,a)] \tag{3}$$

we modify it by applying it to every classifier taking part in the action selection process and weighting the incremental factor by its weight in the perception process as shown in (4).

$$Q(s,a) = Q(s,a) + w_s \times \alpha[r + \gamma \max_a Q(s_{t+1}, *) - Q(s,a)] \tag{4}$$

2.1 Have We Solved the Problem Yet? Unfortunately not

With the introduction of a k-NN approach we add into the algorithm some side effects. These are mainly:

- The action selection mechanism is now more critical due to it is now a collective decision making procedure.
- The learning mechanism is also more complicated since we need to update correctly all relevant neighbors trying to not disturb the information that comes from its adjacency.

But, can we transform these apparent side effects and mayor complexity in something useful for our purposes? Are these new components maybe what we need to improve the learning and decision making procedures? As a first approach we are investigating a feature that has not been investigated rigorously in past approaches. We observe that in most natural experimental learning tasks, the experimenter evaluates and accommodates its learning with relevant information used in the action selection process. Indeed, now we can interpret the action selection mechanism as a machine that generates experiments and expectations with some method and this process must pass all the relevant information about the decision making process to the learning stage as shown in Fig. 3.

2.2 Relevant Variables

We are investigating the interrelation between action-selection and learning by means of three relevant variables:

1. The degree of activation of a classifier.
2. The degree of relevance of a classifier in the action-selection process.
3. The quality of the classifier.

The degree of activation of each classifier is a function of its computed weight w_s when the perception process takes place and is a value in $[0, 1]$ which is computed as:

$$ac_i = \frac{1}{1 + {d_i}^2} \times \frac{1}{decay}, \text{ where the decay factor is a free parameter.} \tag{5}$$

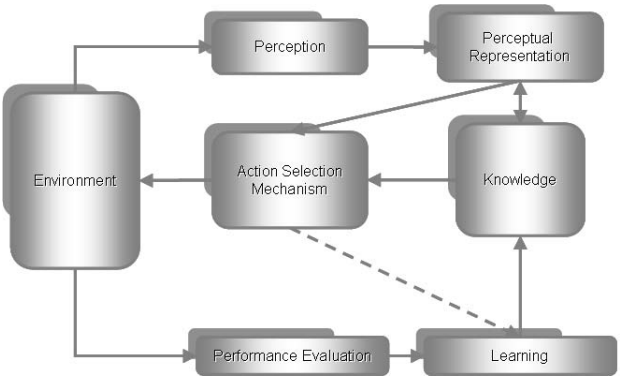


Fig. 3. The Architecture of a Reinforcement Learning Agent where the action selection mechanism affects directly the learning process

The relevance of a classifier can be interpreted as the individual influence of a classifier in a reward prediction process and is a value in $[0, 1]$ which is computed as:

$$r_i = \frac{abs[P(a_{i,1}) - P(a_{i,2})]}{\sum_{j=1}^{n} abs[P(a_{j,1}) - P(a_{j,2})]}, \tag{6}$$

where $P(a_{i,1})$ is the classifier c_i reward prediction for the action selected as the best action as in (2) and $P(a_{i,2})$ is the classifier c_i reward prediction for second ranked action. Thus, the measure r_i is a proportion that evaluates what is the effective contribution of the classifier to the action selection mechanism.

And finally, the quality of a classifier is a progressively estimated measure along all the learning process which refers to how accurate and relevant has been the classifier during the learning process and is a value in $[0, 1]$ which is computed as:

$$q_i = q_i + \alpha \times (1 - r_i) \times \left[\frac{k_i}{\sum k} - q_i \right], \tag{7}$$

where α is a learning rate parameter, r_i is the relevance and k_i is the accuracy of a classifier (an inverse measure of the difference between the predicted and received rewards). Thus any decision making must be based on the degree of activation and the quality of each classifier and the learning procedure must be affected by the quality of the decision and the relevance of each classifier.

2.3 Algorithmic Implementation

The algorithmic implementation of the proposed method is shown in Fig. 4. The update set $[M]$ process consists of updating the reward prediction and all relevant variables for each classifier contained in the match set (k nearest neighbors). The main objective is to update the predicted reward with maximum accuracy.

Initialize Q arbitrarily
Repeat(for each episode)
  s = GetInitialState ()
  Generate the k-NN set $[M]$ out of POPULATION $[cl]$ using s
  Generate the prediction array $P(.)$ using $[M]$
  Select action (a) according to $P(.)$
  Repeat(for n steps)
    Do the selected action (a) and get the next state s'
    Observe the reward R at state s' and the final state flag
    Generate the k-NN set $[M']$ out of POPULATION $[cl]$ using s'
    Generate the prediction array $P(.)$ using $[M']$
    Select action (a') according to $P(.)$
    Update set $[M]$
    $s = s', a = a', M = M'$
  Until s is terminal

Fig. 4. k-NN RL control algorithm

For this purpose we are investigating three modifications to the Q-Learning update rule:

$$Q(s,a,i) = Q(s,a,i) + ac_i \times (1 - r_i) \times \alpha \times [R + \max_a Q(s', *, i) - Q(s', a', i)] \quad (8)$$

$$Q(s,a,i) = Q(s,a,i) + ac_i \times (1 - r_i) \times \alpha \times [R + \max_a P(s', *, *) - Q(s', a', i)] \quad (9)$$

$$Q(s,a,i) = Q(s,a,i) + ac_i \times (1 - r_i) \times \alpha \times [R + T(s', i) - Q(s', a', i)], \quad (10)$$

where T is the linear combination $T = w_q \times \max_a Q(s', *, i) + w_p \times \max_a P(s', *)$

3 Experimental Case Study: Mountain-Car

The presented approach has been applied to several testbed problems, such as the classical mountain-car problem, the pole-cart problem and the acrobot problem and compared to the classical tabular schemes. We present here in detail the results for the mountain-car problem [4, p.214].

Fig. 5 shows the mountain car problem graph and the learning convergence curve for the proposed algorithm in the experiment. Fig. 6 shows the distribution of the classifiers relevance over the state space, it can be seen that the most relevant classifiers are distributed along specific points over the optimal path while irrelevant classifiers are outside the neighborhood of the optimal path. Fig. 7 shows the State-Value surface for different resolutions. Since the developed k-NN scheme acts as a universal continuous function approximator we can test this function approximator with an arbitrary resolution. As can be seen the generalization capability of the proposed perceptual representation (function approximator) is accurate and present some advantages over neural-networks due to its *local* learning capability.

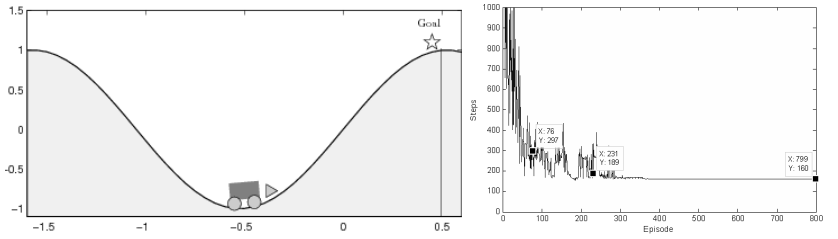


Fig. 5. The classical mountain car problem and experimental convergence curve

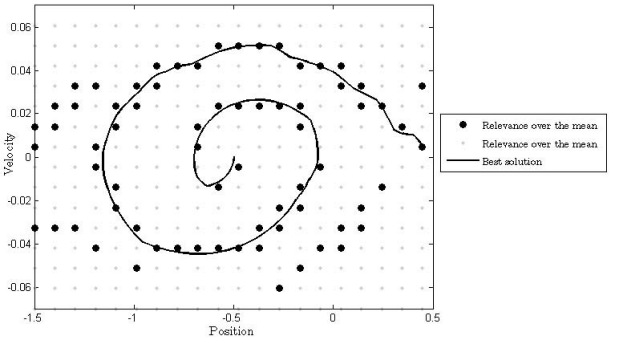


Fig. 6. Distribution of the classifiers Relevance over the state space

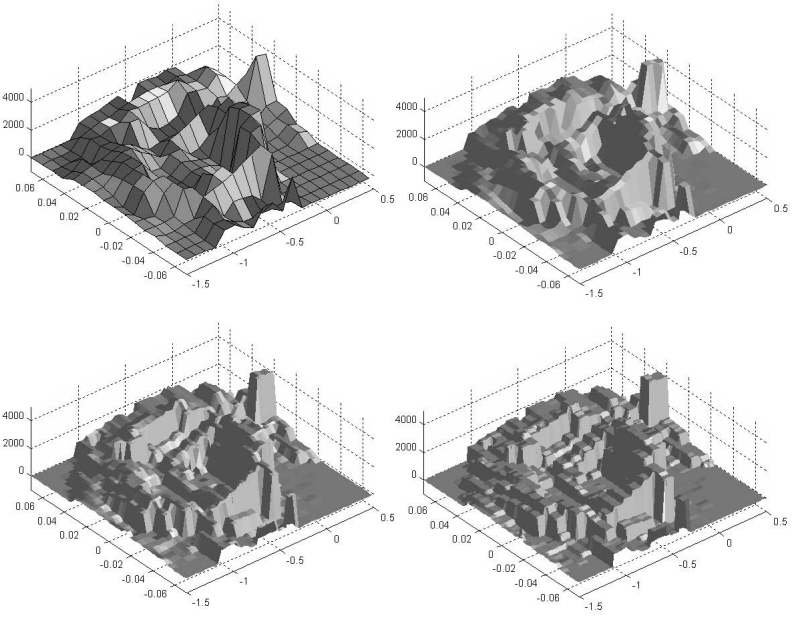


Fig. 7. Learned State-Value surface for different resolutions on the mountain-car

4 Conclusions and Future Work

A solution to the generalization problem in RL algorithms using a k-nearest-neighbor pattern classification (k-NN) has been presented. By the means of the k-NN technique we investigated the effect of multiple decision makers as the main mechanism of perception and action-selection and a sort of back-propagation of its proportional influence in the action-selection process as the factor that moderate the learning of each decision making unit. The effect of collective decision making in the action selection mechanism forces generalization and some interesting characteristics of collective decision making such as the relevance and quality of each decision making unit where analyzed in order to obtain a good learning algorithm.

The experimental results show that the presented k-NN perception scheme for RL problems leads to good generalization and improves the learning rate and robustness of the knowledge acquired by the RL agent.

This line of research opens new problems in the field of optimal classifiers population and collective decision making theory. The final goal of this research is to extend this framework in the context of Evolutionary Online Learning Systems.

References

1. Reinforcement Learning. In: Sutton, R.S (ed.) SECS, vol. 173, Kluwer Academic Publishers (1992). Reprinted from of Machine Learning, vol. 8(3–4) (1992)
2. Sutton, R.S.: Reinforcement learning and artificial intelligence (2006)
3. Sutton, R.S.: Reinforcement learning architectures. In: Proc. Int. Symp. on Neural Information Processing, Kyushu Inst.of Technology, Japan (1992)
4. Sutton, R.S., Barto, A.G.: Reinforcement Learning. An Introduction. MIT Press, Cambridge (1998)
5. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory, IT 13(1), 21–27 (1967)
6. Dudani, S.A.: The distance-weighted k-nearest-neighbor rule. IEEE Transactions on Systems, Man and Cybernetics SMC-6(4), 325–327 (1976)
7. Watkins, C.J., Dayan, P.: Technical note Q-learning. Machine Learning 8, 279 (1992)

On the Estimation of First-Passage Time Densities for a Class of Gauss-Markov Processes*

A.G. Nobile[1], E. Pirozzi[2], and L.M. Ricciardi[2]

[1] Dipartimento di Matematica e Informatica, Università di Salerno, Via Ponte don Melillo, Fisciano (SA), Italy
nobile@unisa.it
[2] Dipartimento di Matematica e Applicazioni, Università di Napoli Federico II, Via Cintia, Napoli, Italy
{enrica.pirozzi, luigi.ricciardi}@unina.it

Abstract. For a class of Gauss-Markov processes the asymptotic behavior of the first passage time (FPT) probability density function (pdf) through certain time-varying boundaries is determined. Computational results for Wiener, Ornstein-Uhlenbeck and Brownian bridge processes are considered to show that the FPT pdf through certain large boundaries exhibits for large times an excellent asymptotic approximation.

1 Introduction

As stressed in previous papers (see, for instance, [5] and references therein), first passage time problems for stochastic continuous-path processes play an essential role in modeling neuronal firing and population extinction. Investigations reported in the literature have essentially proceeded along three main directions: *(i)* to obtain closed-form solutions under suitable assumptions on the involved stochastic processes and on the boundaries; *(ii)* to devise efficient algorithms to reconstruct FPT pdf and to solve the integral equations satisfied by FPT pdf and *(iii)* to analyze the asymptotic behavior of the FPT pdf as boundary or time grow larger. The present contribution goes along similar directions, focusing on a special class of Gauss-Markov processes.

Let $\{Y(t), t \in \mathbb{R}\}$ be the stationary Ornstein-Uhlenbeck (OU) process with zero mean and covariance function $E[Y(s)Y(t)] = \sigma^2\, e^{-\beta\,(t-s)}/(2\,\beta)$ with $s < t$ and $\beta > 0$. Therefore (cf., for instance, [3]):

$$Y(t) = e^{-\beta\,t}\, W\Big(\frac{\sigma^2}{2\,\beta}\, e^{2\,\beta\,t}\Big) \qquad (t \in \mathbb{R}), \tag{1}$$

where $\{W(t), t \geq 0\}$ denotes the standard Wiener process such that $P\{W(0) = 0\} = 1$, $E[W(t)] = 0$ and $E[W(s)\,W(t)] = \min(s,t)$. Furthermore, let $\{Y_1(t), t \geq 0\}$ be the non-stationary OU process with zero mean and covariance function

* Work performed under partial support by MIUR (PRIN 2005) and G.N.C.S.-INdAM.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 146–153, 2007.
© Springer-Verlag Berlin Heidelberg 2007

$E[Y_1(s)Y_1(t)] = \sigma^2 \left(e^{\beta s} - e^{-\beta s}\right) e^{-\beta t}/(2\beta)$ with $s < t$ and $\beta > 0$. Therefore (cf., for instance, [3]):

$$Y_1(t) = e^{-\beta t}\, W\Big[\frac{\sigma^2}{2\beta}\left(e^{2\beta t} - 1\right)\Big] \qquad (t \geq 0). \tag{2}$$

From (1) and (2) one has:

$$Y_1(t) = \sqrt{1 - e^{-2\beta t}}\; Y\Big[\frac{1}{2\beta}\,\ln\Big(e^{2\beta t} - 1\Big)\Big] \qquad (t \geq 0), \tag{3}$$

$$Y(t) = \sqrt{1 + e^{-2\beta t}}\; Y_1\Big[\frac{1}{2\beta}\,\ln\Big(1 + e^{2\beta t}\Big)\Big] \qquad (t \in \mathbb{R}). \tag{4}$$

We now construct a Gaussian process $\{Z(t), t \in T\}$, where T is a continuous parameter set, as follows:

$$Z(t) = m(t) + k(t)\, Y[\varphi(t)] \qquad (t \in T), \tag{5}$$

where *(a)* $m(t)$ is continuous in T; *(b)* $\varphi(t) : T \to \mathbb{R}$ is a continuous and monotonically increasing function; *(c)* $k(t)$ is a continuous and not vanishing function in the interior of T. Recalling (4), from (5) one has:

$$Z(t) = m(t) + k_1(t)\, Y_1\big[\varphi_1(t)\big] \qquad (t \in T), \tag{6}$$

where we have set:

$$k_1(t) = k(t)\,\sqrt{1 + e^{-2\beta\varphi(t)}}, \qquad \varphi_1(t) = \frac{1}{2\beta}\,\ln\big(1 + e^{2\beta\varphi(t)}\big). \tag{7}$$

By virtue of *(b)* and *(c)*, from (7) we note that $\varphi_1(t) : T \to [0, +\infty)$ is a continuous and monotonically increasing function and $k_1(t)$ is a continuous and not vanishing function in the interior of T. Making use of (1) in (5), or equivalently making use of (2) in (6), one also obtains:

$$Z(t) = m(t) + h_2(t)\, W[r(t)] \qquad (t \in T), \tag{8}$$

with

$$\begin{aligned} h_1(t) &:= \frac{\sigma^2}{2\beta}\, k(t)\, e^{\beta\varphi(t)} = \frac{\sigma^2}{2\beta}\, k_1(t)\left[e^{\beta\varphi_1(t)} - e^{-\beta\varphi_1(t)}\right] \\ h_2(t) &:= k(t)\, e^{-\beta\varphi(t)} = k_1(t)\, e^{-\beta\varphi_1(t)} \qquad\qquad (t \in T). \\ r(t) &:= \frac{h_1(t)}{h_2(t)} = \frac{\sigma^2}{2\beta}\, e^{2\beta\varphi(t)} = \frac{\sigma^2}{2\beta}\left[e^{2\beta\varphi_1(t)} - 1\right] \end{aligned} \tag{9}$$

Hence, $\{Z(t), t \in T\}$ is a real continuous Gauss-Markov process with mean $E[Z(t)] = m(t)$ and covariance

$$c(s,t) = E\{[Z(s) - m(s)]\,[Z(t) - m(t)]\} = h_1(s)\, h_2(t) \qquad (s < t).$$

The transition pdf $f_{_Z}(x,t|z,\tau)$ of the Gauss-Markov process $Z(t)$ is a normal density characterized respectively by mean and variance:

$$E[Z(t) \mid Z(\tau)=z] = m(t) + \frac{h_2(t)}{h_2(\tau)}\,\big[z - m(\tau)\big]$$
$$\mathrm{Var}[Z(t) \mid Z(\tau)=z] = h_2(t)\Big[h_1(t) - \frac{h_2(t)}{h_2(\tau)}h_1(\tau)\Big]$$

for $t,\tau \in T$ and $\tau < t$. The infinitesimal moments of $Z(t)$ are given by

$$A_1(x,t) = m'(t) + [x - m(t)]\,\frac{h_2'(t)}{h_2(t)}, \qquad A_2(t) = h_2^2(t)\,r'(t),$$

the prime denoting derivative with respect to the argument.

We shall now focus our attention on the random variable

$$\mathcal{T}_z = \inf_{t\geq\tau}\big\{t : Z(t) > S(t)\big\}, \qquad Z(\tau)=z<S(\tau), \quad \tau,t\in T \tag{10}$$

that represents the first-passage time of $Z(t)$ from $Z(\tau)=z$ to the supposed continuous boundary $S(t)$. Making use of (8), the FPT pdf g_Z of $Z(t)$ can be obtained from the FPT pdf g_W of the Wiener process $W(t)$ as:

$$g_{_Z}[S(t),t|z,\tau] := \frac{\partial}{\partial t}P\big(\mathcal{T}_z<t\big) = \frac{dr(t)}{dt}\,g_{_W}\left\{\frac{S(t)-m(t)}{h_2(t)}, r(t)\,\Big|\,\frac{z-m(\tau)}{h_2(\tau)}, r(\tau)\right\} \tag{11}$$

for $z<S(\tau)$. By virtue of (5) and (6), g_Z can be also expressed in terms of the FPT pdf g_Y of the stationary OU process $Y(t)$ and in terms of the FPT pdf $g_{_{Y_1}}$ of the non-stationary OU process $Y_1(t)$ as follows:

$$\begin{aligned} g_{_Z}[S(t),t \mid z,\tau] &= \frac{d\varphi(t)}{dt}\,g_{_Y}\Big[\frac{S(t)-m(t)}{k(t)}, \varphi(t)\,\Big|\,\frac{z-m(\tau)}{k(\tau)}, \varphi(\tau)\Big] \\ &= \frac{d\varphi_1(t)}{dt}\,g_{_{Y_1}}\Big[\frac{S(t)-m(t)}{k_1(t)}, \varphi_1(t)\,\Big|\,\frac{z-m(\tau)}{k_1(\tau)}, \varphi_1(\tau)\Big], \qquad z<S(\tau). \end{aligned} \tag{12}$$

As proved in [1], if $S(t), m(t), h_1(t), h_2(t)$ are $C^1(T)$ functions, the FPT pdf of the Gauss-Markov process $Z(t)$ satisfies the following nonsingular second-kind Volterra integral equation

$$g_{_Z}[S(t),t|z,\tau] = -2\,\Psi[S(t),t|z,\tau] + 2\int_\tau^t g_{_Z}[S(\vartheta),\vartheta|z,\tau]\,\Psi[S(t),t \mid S(\vartheta),\vartheta]\,d\vartheta,$$

for $z<S(\tau)$, where

$$\begin{aligned} \Psi[S(t),t \mid z,\vartheta] = \Big\{ &\frac{S'(t)-m'(t)}{2} - \frac{S(t)-m(t)}{2}\,\frac{h_1'(t)h_2(\vartheta)-h_2'(t)h_1(\vartheta)}{h_1(t)h_2(\vartheta)-h_2(t)h_1(\vartheta)} \\ &- \frac{z-m(\vartheta)}{2}\,\frac{h_2'(t)h_1(t)-h_2(t)h_1'(t)}{h_1(t)h_2(\vartheta)-h_2(t)h_1(\vartheta)}\Big\}\,f_{_Z}[S(t),t \mid z,\vartheta]. \end{aligned} \tag{13}$$

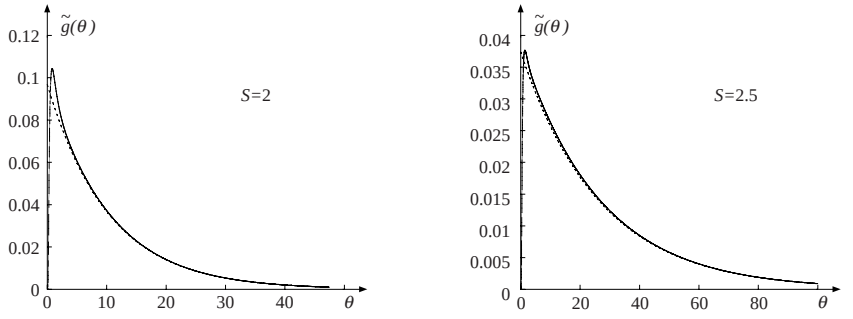


Fig. 1. For the non-stationary OU process with $\beta = 1$ and $\sigma^2 = 2$, $\tilde{g}(\vartheta) = \tilde{g}_{Y_1}(S, \vartheta|0, 0)$ (solid line) is compared with the exponential density $\lambda \exp(-\lambda\,\vartheta)$ (dot line), with λ estimated as $\widehat{\lambda} = 0.09688$ for $S = 2$ and $\widehat{\lambda} = 0.03741$ for $S = 2.5$

In [1] a computationally simple, speedy and accurate method based on the repeated Simpson rule is proposed to construct FPT pdf for Gauss-Markov processes through time-dependent boundaries. In the following we shall denote by $\tilde{g}(t_k)$ the numerical evaluation of $g_Z[S(t_k), t_k|z, \tau]$ at times $t_k = \tau + k\,p$ $(k = 1, 2, \ldots)$, where $p > 0$ is the time discretization step.

2 Asymptotic Behavior

We now assume that the boundary of the Gauss-Markov process $Z(t)$ is

$$\begin{aligned} S_1(t) &:= m(t) + S\,k_1(t) = m(t) + S\,k(t)\,\sqrt{1 + e^{-2\,\beta\,\varphi(t)}} \\ &= m(t) + S\,h_2(t)\,\sqrt{1 + 2\,\beta\,r(t)/\sigma^2} \qquad (t \in T, S \in \mathbb{R}), \end{aligned} \tag{14}$$

where use of (7) and (9) has been made. Hence, setting $S(t) = S_1(t)$ in (12), one obtains:

$$g_Z[S_1(t), t \mid z, \tau] = \frac{d\varphi_1(t)}{dt}\, g_{Y_1}\Big[S, \varphi_1(t)\,\Big|\,\frac{z - m(\tau)}{k_1(\tau)}, \varphi_1(\tau)\Big], \quad z < S_1(\tau), \tag{15}$$

being the first passage time of $Z(t)$ through the boundary $S_1(t)$ a sure event if

$$\lim_{t \to \sup T} \varphi_1(t) = +\infty. \tag{16}$$

As shown in [2] and [4], the FPT pdf of the non-stationary OU process $Y_1(t)$ through constant large S and large times is susceptible of an excellent exponential approximation

$$g_{Y_1}(S, \vartheta \mid y, \vartheta_0) \simeq \lambda\, e^{-\lambda\,(\vartheta - \vartheta_0)} \qquad (\vartheta > \vartheta_0 \geq 0,\ \lambda > 0), \tag{17}$$

where the least squares estimate of λ is:

$$\widehat{\lambda} = -\,\frac{\sum_{k=1}^{N} (\vartheta_k - \vartheta_0)\,\ln[1 - \tilde{G}_{Y_1}(\vartheta_k)]}{\sum_{k=1}^{N} (\vartheta_k - \vartheta_0)^2}, \tag{18}$$

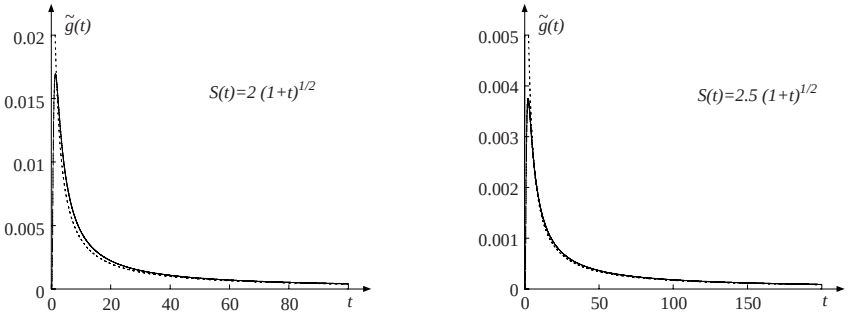


Fig. 2. For the standard Wiener process ($\mu = 0$, $\omega^2 = 1$), $\tilde{g}(t) = \tilde{g}_{Z}(S\sqrt{1+t}, t \mid 0, 0)$ (solid line) is compared with the asymptotic density $\widehat{g}(t)$ (dot line) given in (23) for the same choices of λ and S of Fig.1

where $\vartheta_k = \vartheta_0 + k\,p$ and $\tilde{G}_{Y_1}(\vartheta_k)$ denotes the evaluated FPT distribution function. Fig. 1 shows the evaluated FPT density $\tilde{g}(\vartheta) = \tilde{g}_{Y_1}(S, \vartheta|0, 0)$ with $S = 2$ and $S = 2.5$ and the exponential density $\lambda \exp(-\lambda\,\vartheta)$, with λ estimated by means of (18), for the non-stationary OU process with $\beta = 1$ and $\sigma^2 = 2$.

The asymptotic behavior of g_{Y_1} can thus be used to obtain asymptotic properties of g_{Z}. Indeed, by virtue of (17), under the assumption (16) the following asymptotic behavior holds for large S and large times:

$$g_{Z}[S_1(t), t \mid z, \tau] \simeq \widehat{g}(t) := \frac{d\varphi_1(t)}{dt}\,\lambda\, e^{-\lambda\,[\varphi_1(t)-\varphi_1(\tau)]} \qquad (\tau, t \in T,\ t > \tau). \quad (19)$$

Recalling (7) and (9), $\widehat{g}(t)$ can be also written as:

$$\widehat{g}(t) = \frac{d\varphi(t)}{dt}\,\frac{\lambda\, e^{2\,\beta\,\varphi(t)}\left[1 + e^{2\,\beta\,\varphi(\tau)}\right]^{\lambda/(2\,\beta)}}{\left[1 + e^{2\,\beta\,\varphi(t)}\right]^{1+\lambda/(2\,\beta)}} = \frac{\lambda}{\sigma^2}\,\frac{dr(t)}{dt}\,\frac{\left[1 + 2\beta\, r(\tau)/\sigma^2\right]^{\lambda/(2\,\beta)}}{\left[1 + 2\beta\, r(t)/\sigma^2\right]^{1+\lambda/(2\,\beta)}}$$
$$(\tau, t \in T,\ t > \tau). \quad (20)$$

The goodness of approximation (19) and (20), with λ estimated via (18), has been confirmed by our computations for a wide class of Gauss-Markov processes in the presence of special time-varying boundaries.

2.1 Wiener Process

Let $Z(t)$ be the Wiener process in the time domain $T = [0, +\infty)$ such that

$$Z(t) = \mu\, t + W(\omega^2\, t) \qquad (\mu \in \mathbb{R}, \omega > 0). \quad (21)$$

Since $m(t) = \mu\, t$, $h_2(t) = 1$ and $r(t) = \omega^2\, t$, setting $\beta = 1$ and $\sigma^2 = 2$ from (14) one has

$$S_1(t) = m(t) + S\, h_2(t)\,\sqrt{1 + 2\,\beta\, r(t)/\sigma^2} = \mu\, t + S\,\sqrt{1 + \omega^2\, t} \qquad (t \geq 0). \quad (22)$$

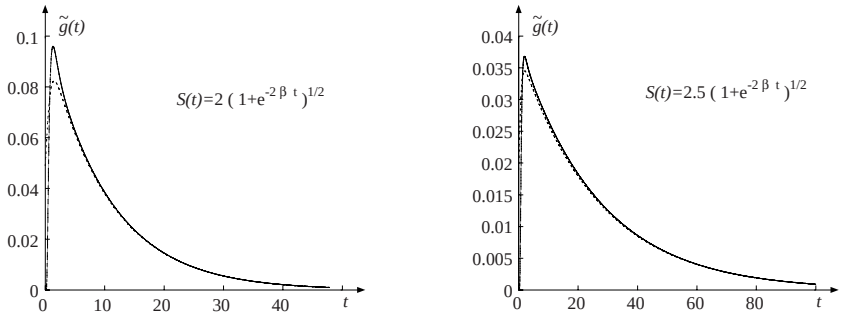


Fig. 3. The FPT pdf $\tilde{g}(t) = \tilde{g}_{Y}(S\sqrt{1+e^{-2\beta t}}, t \mid 0,0)$ (solid line) for the stationary OU process ($\beta = 1$, $\sigma^2 = 2$) is compared with the asymptotic density $\widehat{g}(t)$ (dot line) given in (24) for the same choices of λ and S of Fig.1

Hence, recalling (20), the FPT pdf of the Wiener process through the boundary $S_1(t)$ exhibits for large S and large times the following asymptotic behavior:

$$g_{Z}\left[\mu t + S\sqrt{1+\omega^2 t},\, t \mid z, \tau\right] \simeq \frac{\lambda\,\omega^2}{2}\,\frac{\left(1+\omega^2\,\tau\right)^{\lambda/2}}{\left(1+\omega^2\,t\right)^{1+\lambda/2}} \equiv \widehat{g}(t), \qquad (23)$$

with $0 \le \tau < t < +\infty$. The evaluated FPT density $\tilde{g}(t) = \tilde{g}_{Z}(S\sqrt{1+t}, t \mid 0,0)$ is shown in Fig. 2 for the standard Wiener process ($\mu = 0$, $\omega^2 = 1$) with $S = 2$ and $S = 2.5$ and the asymptotic density $\widehat{g}(t)$ given in (23) with λ obtained by means of (18).

2.2 Stationary OU Process

Let $Z(t) = Y(t)$ be the stationary OU process defined in (1) with time domain $T = \mathbb{R}$. Since $m(t) = 0$, $h_2(t) = e^{-\beta\,t}$ and $r(t) = \sigma^2 e^{2\,\beta\,t}/(2\,\beta)$, from (14) one obtains:

$$S_1(t) = m(t) + S\,h_2(t)\,\sqrt{1+2\,\beta\,r(t)/\sigma^2} = S\,\sqrt{1+e^{-2\,\beta\,t}}.$$

Hence, recalling (20), the FPT pdf of $Y(t)$ through the boundary $S_1(t)$ exhibits for large S and large times the following asymptotic behavior:

$$g_{Y}\left[S\sqrt{1+e^{-2\,\beta\,t}}, t \mid z, \tau\right] \simeq \frac{\lambda\,e^{2\,\beta\,t}\left(1+e^{2\,\beta\,\tau}\right)^{\lambda/(2\,\beta)}}{\left(1+e^{2\,\beta\,t}\right)^{1+\lambda/(2\,\beta)}} \equiv \widehat{g}(t) \qquad (24)$$

with $-\infty < \tau < t < +\infty$. Fig. 3 shows $\tilde{g}(t) = \tilde{g}_{Y}(S\sqrt{1+e^{-2\,\beta\,t}}, t \mid 0,0)$ for the stationary OU process ($\beta = 1$, $\sigma^2 = 2$) with $S = 2$ and $S = 2.5$ and the asymptotic density $\widehat{g}(t)$ given in (24) with λ obtained by means of (18).

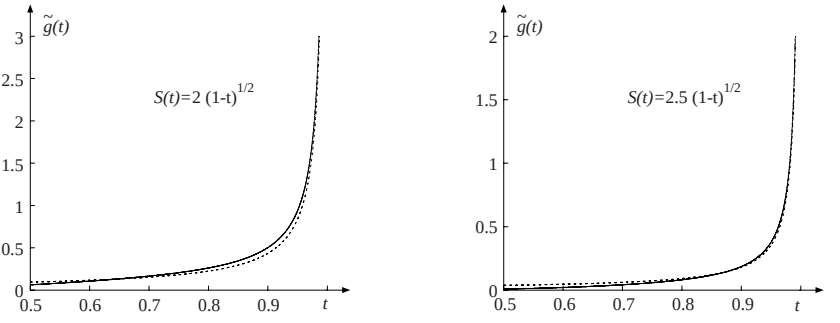


Fig. 4. For the Brownian bridge $\tilde{g}(t) = \tilde{g}_{Z}(S\sqrt{1-t}, t \mid 0, 0)$ (solid line) is compared with the asymptotic density $\widehat{g}(t)$ (dot line) given in (26) for the same choices of λ and S of Fig.1.

2.3 Brownian Bridge

We now consider the Brownian bridge process with time domain $T = [0, 1)$, defined as:

$$Z(t) = (1-t)\, W\Big(\frac{t}{1-t}\Big). \tag{25}$$

In this case one has $m(t) = 0$, $h_2(t) = 1 - t$ and $r(t) = t/(1-t)$. Setting $\beta = 1$ and $\sigma^2 = 2$, from (14) one has:

$$S_1(t) = m(t) + S\, h_2(t)\, \sqrt{1 + 2\,\beta\, r(t)/\sigma^2} = S\,\sqrt{1-t}.$$

Hence, for large S and large times the following asymptotic behavior holds:

$$g_{Z}\Big[S\sqrt{1-t},\, t \mid z, \tau\Big] \simeq \frac{\lambda}{2}\, \frac{(1-t)^{-1+\lambda/2}}{(1-\tau)^{\lambda/2}} \equiv \widehat{g}(t) \qquad (0 \le \tau < t < 1). \tag{26}$$

Fig. 4 shows the evaluated FPT density $\tilde{g}(t) = \tilde{g}_{Z}(S\sqrt{1-t}, t \mid 0, 0)$ for the Brownian bridge with $S = 2$ and $S = 2.5$ and the density $\widehat{g}(t)$ given in (26).

2.4 A Transformation of the Non-stationary OU Process

Let $Z(t)$ be the following Gauss-Markov process with time domain $T = [0, 1)$

$$Z(t) = (1-t)\, Y_1\Big(\frac{t}{1-t}\Big). \tag{27}$$

Recalling (6), in this case one has $k_1(t) = 1 - t$ and $\varphi_1(t) = t/(1-t)$, so that $Z(t)$ is a Gauss-Markov process with $m(t) = 0$ and

$$h_1(t) = \frac{\sigma^2}{2\,\beta}\,(1-t)\,\Big[\exp\Big\{\frac{\beta\, t}{1-t}\Big\} - \exp\Big\{-\frac{\beta\, t}{1-t}\Big\}\Big], \quad h_2(t) = (1-t)\,\exp\Big\{-\frac{\beta\, t}{1-t}\Big\}.$$

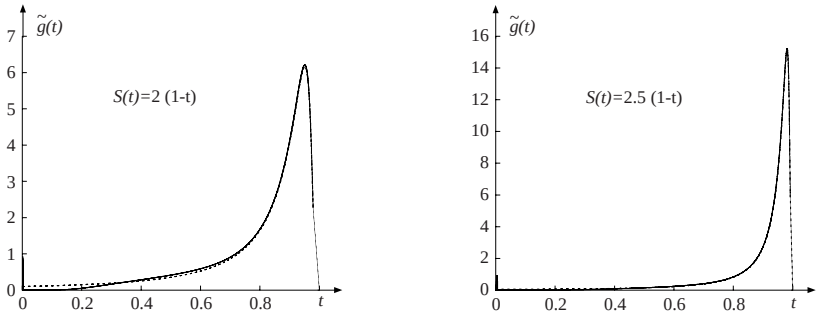


Fig. 5. For the process $Z(t)$ given in (27) with $\beta = 1$ and $\sigma^2 = 2$, $\tilde{g}(t) = \tilde{g}_{Y}(S(1-t), t \mid 0, 0)$ (solid line) is compared with the asymptotic density $\widehat{g}(t)$ (dot line) given in (28) for the same choices of λ and S of Fig.1.

The infinitesimal moments of the process $Z(t)$ defined in (27) are $A_1(x,t) = -x\,(1+\beta-t)/(1-t)^2$ and $A_2(t) = \sigma^2$ $(0 \le t < 1)$, so that as $\beta \to 0$ and $\sigma^2 = 1$ the process $Z(t)$ becomes the Brownian bridge process (25). From (14) one has:

$$S_1(t) = m(t) + S\,k_1(t) = S\,(1-t),$$

so that for large S and large times the following asymptotic behavior holds:

$$g_{Z}\Big[S\,(1-t),\, t \mid z, \tau\Big] \simeq \frac{\lambda}{(1-t)^2}\,\exp\Big\{-\lambda\Big[\frac{t}{1-t} - \frac{\tau}{1-\tau}\Big]\Big\} \equiv \widehat{g}(t) \qquad (28)$$

with $0 \le \tau < t < 1$. Fig. 5 shows the evaluated FPT density $\tilde{g}(t) = \tilde{g}_{Z}(S\,(1-t), t \mid 0, 0)$ for the process (27) ($\beta = 1$, $\sigma^2 = 2$) with $S = 2$ and $S = 2.5$ and the asymptotic density $\widehat{g}(t)$ given in (28).

A detailed proof of the asymptotic result (19) and (20) for the Gauss-Markov processes as well as some their extensions will be provided in future works.

References

1. Di Nardo, E., Nobile, A.G., Pirozzi, E., Ricciardi, L.M.: A computational approach to first-passage-time problems for Gauss-Markov processes. Adv. Appl. Prob. 33, 453–482 (2001)
2. Giorno, V., Nobile, A.G., Ricciardi, L.M.: On the asymptotic behaviour of first–passage–time densities for one–dimensional diffusion processes and varying boundaries. Adv. Appl. Prob. 22, 883–914 (1990)
3. Larson, H.J., Shubert, B.O.: Probabilistic models in engineering sciences. In: Random variables and stochastic processes, vol. 1, John Wiley & Sons, New York (1979)
4. Nobile, A.G., Ricciardi, L.M., Sacerdote, L.: Exponential trends of Ornstein-Uhlenbeck first-passage-time densities. J. Appl. Prob. 22, 360–369 (1985)
5. Ricciardi, L.M., Di Crescenzo, A., Giorno, V., Nobile, A.G.: An outline of theoretical and algorithmic approaches to first passage time problems with applications to biological modeling. Math. Japonica 50, 247–322 (1999)

Simulation of Myosin II Dynamics Modeled by a Pulsating Ratchet with Double-Well Potentials*

A. Buonocore[1], L. Caputo[2], E. Pirozzi[1], and L.M. Ricciardi[1]

[1] Dipartimento di Matematica e Applicazioni,
Università di Napoli Federico II, Via Cintia, Napoli, Italy
{anibuono,epirozzi,lmr}@unina.it
[2] Dipartimento di Matematica,
Università di Torino, Via Carlo Alberto, Torino, Italy
luigia.caputo@unito.it

Abstract. A detailed analysis is carried to show that a Brownian motor driven by suitable double-well potentials randomly alternating with realistically chosen parameters can account for the experimentally evaluated relevant quantities.

1 Introduction

As is well known, the sliding of Myosin II ("myosin", in the sequel) head along an actin filament, during a chemical ATP cycle, plays a key role in the muscle contraction process. Although molecular motors of various kinds have been proposed during the last decade to try to explain the main dynamical features of proteins (see, for instance, [9] and references therein), it is fair to claim that the detailed biological mechanism responsible for converting chemical energy into directional motion is still rather obscure. Thanks to highly sophisticated novel methods, numerous and reliable experimental data on actin-myosin interaction have been made available over the past few years ([7]). These have been the object of various studies and fitting trials (see, for instance, [3]); in addition, they have stimulated the search for phenomenological models aiming at quantitative explanations and predictions. Proposed models are mainly centered on ratchets or Brownian motors ([11]); these are not restricted to myosin dynamics, but are also appropriate for a wider range of biophysical phenomena. An important characteristic feature of ratchets is their energetic efficiency, quantitatively indicating how much they are able to convert fluctuations into useful work, e.g., to contrast the effect of an external force ([10]). Generally, flashing ratchets lead to energetic efficiency values smaller than those implied by the experimental results ([1]). However, recent investigations have demonstrated that certain ratchet models based on alternating potentials having identical spatial periods but being mutually shifted by half a period may exhibit higher efficiencies ([8], [6]).

* Work performed under partial support by MIUR (PRIN 2005) and G.N.C.S. - INdAM.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 154–162, 2007.
© Springer-Verlag Berlin Heidelberg 2007

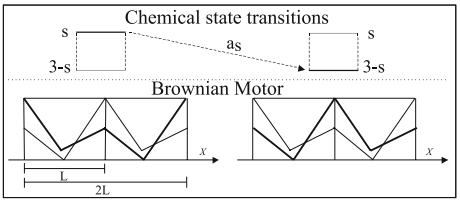


Fig. 1. Sketch of the Brownian motor. On top, state transitions induced by a chemical reaction are indicated. In the bottom, the corresponding active double-well potential (thick lines), and the previous potential (thin lines) are drawn.

In [4] e in [5] we invoked a Brownian motor to model the motion of particles in an asymmetric, single-well, periodic potential (whose period equals twice the distance, say L, between successive monomers of the actin filament) undergoing Poisson distributed half-period shifts. The Brownian motor operates in agreement with the assumption of random Poisson-distributed switching times between the two chemical states modeled by the $2L$-periodic potentials. Relevant quantities, i.e. probability currents, effective driving force([11]), stall force, power and efficiency of the motor, were explicitly obtained as averages of certain functions of the random variable representing the particle position. Use of the notion of driving effective potential was made to bridge our model with our previous works involving washboard potentials. In particular, we could prove that if transitions between the chemical states occur with equal rates, the effective driving potential takes the form of an L-periodic, tilted potential, i.e. the washboard potential originally postulated in [2] on purely phenomenological grounds.

In order to come up with an increased efficiency of the Brownian motor proposed in [4], in the present contribution, inspired by [8] and with the aim to make our formulation richer, we adopt the above-mentioned double-well potential hypothesis and introduce a space parameter to regulate the distance between the potential wells. Thus doing, also in this case we are able to quantify the motor features. In addition, our computations definitely indicate the desirable increased values of the motor efficiency.

2 The Model

Under the assumptions that the motor step equals the distance L between successive actin monomers (roughly 5.5 nm), along the lines of [4], here we propose a pulsating ratchet model based on the following Langevin-equation:

$$\dot{x}(t) = -\frac{U_S'(x) - F_e}{\rho} + \sqrt{\frac{2k_{\mathrm{B}}T}{\rho}}\Lambda(t), \tag{1}$$

where F_e, ρ and T are parameters representing a possible externally applied force, the viscous drag coefficient and the absolute temperature, respectively.

By $\Lambda(t)$ we denote a standard Gaussian white noise and by k_{B} the Boltzmann constant. Finally, $U_S(x)$ denotes the above mentioned $2L$-periodic *double-well* potentials, where S is a random variable, with spectrum $\{1, 2\}$, specifying the current chemical state and related to the fluctuation source. For $s \in \{1, 2\}$ the transition from chemical state s to chemical state $3 - s$ is assumed to occur according to a Poisson process of rate a_s. Setting $a = a_1 + a_2$, it is easy to show that $P_1 := P(S = 1) = a_2/a$ and $P_2 := P(S = 2) = a_1/a$. Moreover,

$$U_s(x + L) = U_{3-s}(x), \quad \text{for } x \in [0, L] \text{ and } s \in \{1, 2\}. \tag{2}$$

Figure 1 sketches this dynamics for the case of saw-tooth potentials U_1 and

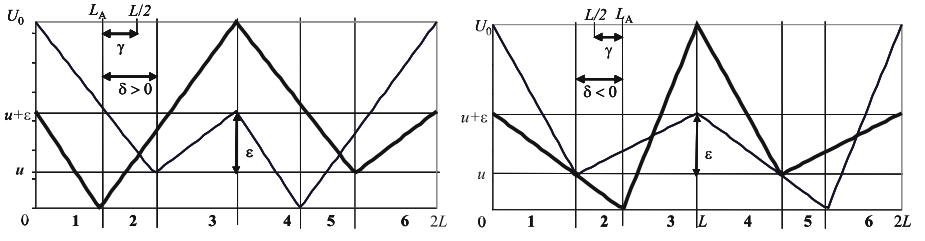


Fig. 2. Couples of alternating double-well saw-tooth potentials, with $2L$ periodicity and spatially L -shifted from one other, are plotted ($U_1(x)$ thick line, $U_2(x)$ thin line) for different choices of parameters γ and δ. The $(0, 2L)$ interval is partitionated into 6 subintervals in each of which the derivative of at least one of two potentials changes.

U_2. Here, we postulate that the first well of the potential $U_1(x)$ is located at $L_A = L/2+\gamma$ (with$-L/2 < \gamma < L/2$) while the second well is $L+\delta$ (with $\delta \in \mathbb{R}$) apart from the first one. In Figure 2 two example of potentials configurations (for $\delta > 0$ on the left and for $\delta < 0$ on the right) witness the role of γ and δ in determining the locations of potential wells. Here, U_0 denotes the maximum of the potentials, whereas u e $u + \varepsilon$ represent the depths of the potential wells, respectively.

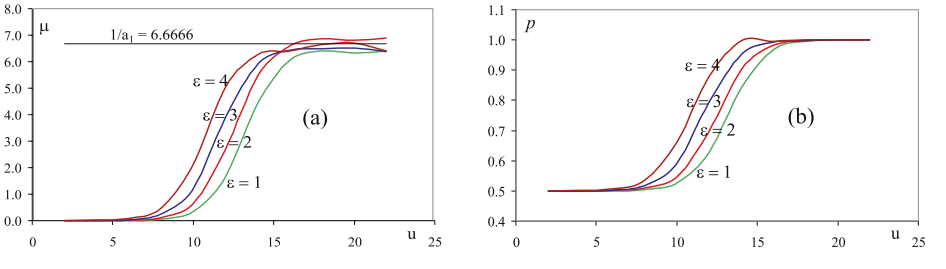


Fig. 3. Plots of mean first-exit time (a) and of forward step probability (b) versus u for some small values of ε

The Fokker-Planck formulation, modeling the evolution of an ensemble of identical particles each of which is subject to the same dynamics (1), is

$$\begin{cases} \frac{\partial p_1(x,t)}{\partial t} = -a_1 p_1(x,t) + a_2 p_2(x,t) + \frac{\partial}{\partial x}\left[\frac{U_1'(x)-F_e}{\rho} p_1(x,t)\right] + D\frac{\partial^2}{\partial x^2} p_1(x,t) \\ \frac{\partial p_2(x,t)}{\partial t} = +a_1 p_1(x,t) - a_2 p_2(x,t) + \frac{\partial}{\partial x}\left[\frac{U_2'(x)-F_e}{\rho} p_2(x,t)\right] + D\frac{\partial^2}{\partial x^2} p_2(x,t), \end{cases} \tag{3}$$

where $D = k_{\mathrm{B}}T/\rho$ denotes the diffusion constant. When solving equations (3), positions x are confined to interval $(0, 2L)$, meaning that the ($2L$-periodic) obtained solution must take into account the contributions of all modulo $2L$ congruous abscissae. Then, for each $x \in (0, 2L)$ and for each positive or negative integer k, the product $p_s(x,t)dx$, with $s \in \{1, 2\}$, yields the fraction of particles that at time t are in the neighborhood of each position $x + 2kL$, jointly with occurrence of event $\{S = s\}$. Note that in such reduced dynamics the existence of a stationary regime is suggested by physical considerations (see, for instance, [9]). Hereafter, the solution of (3) in the stationary regime will be denoted by $p_s(x)$, while

$$J_s(x) := -D\left[\frac{U_s'(x) - F_e}{\rho D} p_s(x) + p_s'(x)\right], \tag{4}$$

will denote the probability current (i.e. the particles flux) in s-state. Note that for $s \in \{1, 2\}$, both $p_s(x)$ and $J_s(x)$ depend on all previously introduced parameters. When necessary, the explicit dependence of $p_s(x)$ and $J_s(x)$ on certain parame-

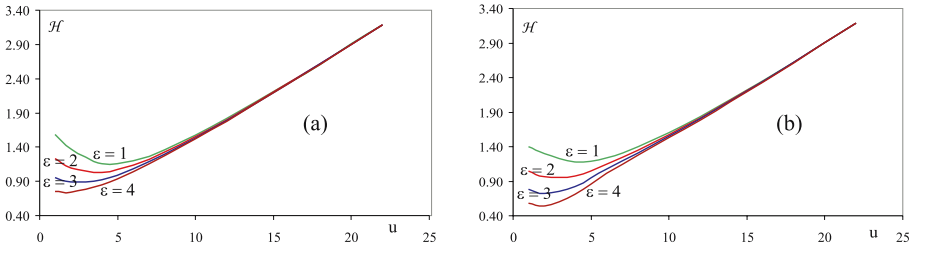


Fig. 4. Plots of mean energy consumption $\mathcal{H}$ versus u for some small values of ε in the case $F_e = 0$ (a) and $F_e = -2$ pN (b)

ters will be indicated. The probability density function of X is then immediately obtained as $p(x) = p_1(x) + p_2(x)$. In the stationary regime the probability conservation law insures that the total flux $J(x) = J_1(x) + J_2(x)$ is a constant, say J. In the sequel we shall make use of the definition

$$E\left[g(X,S)\right] = \sum_{s=1}^{2} \int_0^{2L} g(x,s) p_s(x)\, dx \tag{5}$$

expressing the average of a measurable function g of the random vector (X, S), and assume that $p(x)$ is continuous in $[0, 2L]$ while $p'(x)$ admits a finite number

of first-kind discontinuities. Proposition 1 of [5] is valid also in the present model. Hence, we can adopt the analytical descriptions and the similar evaluation technique of the quantities that are relevant in this context, as therein described. Namely, let F_i, $\mathcal{H}$ and η be the stall force, the effective driving force, the average energy consumption in unit of time and the motor efficiency, respectively. One has

$$J = \frac{\rho^{-1}}{2L}\left\{F_e + E\left[-U_S'(X)\right]\right\}; \quad F_i = -\frac{1}{2L}E\left[\frac{U_S'(X)}{p(X)}\right]; \tag{6}$$

$$\mathcal{H} = E\left[a_S \Delta U_S(X)\right]; \quad \eta = \frac{\left\{F_e + E\left[-U_S'(X)\right]\right\} \cdot F_e}{E\left[a_S \Delta U_S(X)\right]} \cdot \rho^{-1}. \tag{7}$$

Note that the stall force F_{st} of the Brownian motor is the fixed point of $E\left[U_S'(X)\right]$ considered as a function of F_e.

3 Computational Results

Hereafter, unless differently stated, we shall take throughout $F_e = 0$, $a_1 = a_2 = 1.5 \cdot 10^{-1}$/ms, $L = 5.5$ nm, $\rho = 9 \cdot 10^{-5}$ pN·ms/nm, $T = 293$ °K and $U_0 = 32$ $k_{\mathrm{B}}T$. Furthermore, we shall tacitly assume that lengths are expressed in nm, times in ms, forces in pN, energies in $k_{\mathrm{B}}T$ units and $\mathcal{H}$ in $k_{\mathrm{B}}T$/ms. In the case of saw-tooth potentials, (see Figure 2), eqs. (3) can be analytically solved. Having determined p_1 and p_2, the other relevant quantities are then obtained from (6) and (7).

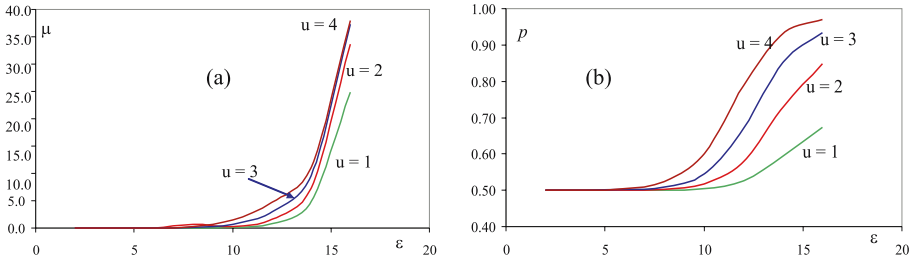


Fig. 5. Plots of mean first-exit time (a) and of forward step probability (b) versus ε for some small values of u

We now describe the mean first-exit time μ, the probability of forward step p, the mean energy consumption per ms $\mathcal{H}$ with respect to u for various small values of ε and with $\gamma = \delta = 0$, i.e. when the two potential wells are located at $L/2$ and $3L/2$, respectively. In Figure 3 the first-exit mean time (a) and the step forward probability (b) are plotted as function of u. It is seen that the first-exit mean time μ asymptotically approaches, or even overcomes, the single state transition mean time $1/a_1$. Moreover, it is evident that as u decreases the probability of backward exits increases. At the same time the overall mean

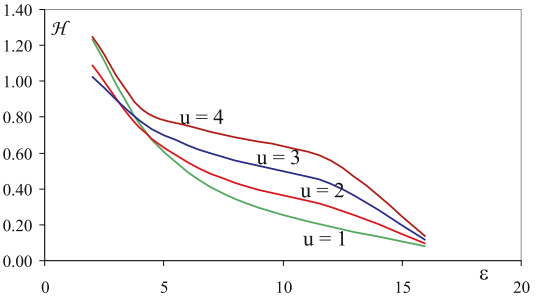


Fig. 6. Plots of mean energy consumption $\mathcal{H}$ versus ε for some small values of u in the case $F_e = 0$

first-exit time decreases due to the higher frequency of backward exits. For u of about 14 $k_{\mathrm{B}}T$, the obtained pairs (μ, p) fall well within the range of the experimentally observed values irrespectively of the chosen ε. However, as shown by Figure 4, the corresponding average energy consumption per ms would attain unrealistic values larger than 2 $k_{\mathrm{B}}T$ independently of the load condition ($F_e = 0$ or $F_e = -2$pN in the Figure). Next, we look at μ and p versus ε for certain small values of u and for $\gamma = \delta = 0$. Figure 5 shows (a) that the first-exit mean time grows very rapidly with ε. The reason is that as ε increases, not all state transitions yield a forward step while, in addition, the depth of the potential responsible of the backward steps increases. As shown in (b), the probability of a forward exit is an increasing function of ε and of μ. As for the mean energy consumption per ms $\mathcal{H}$, in the absence of external load this is seen to decrease with ε (see Figure 6). For $\varepsilon \geq 6\ k_{\mathrm{B}}T$, it increases with u, though remaining below 1 $k_{\mathrm{B}}T$/ms. A similar qualitative behavior of $\mathcal{H}$ has been observed under different load conditions as well, and even smaller values of $\mathcal{H}$ have been observed for loads between 1 and 3 pN. To summarize, the results of our numerical analysis, sketched in the foregoing, indicate that in order for the present model to lead to realistic values one should take $\varepsilon > 10\ k_{\mathrm{B}}T$ (which yields $p >> 0.5$) and $\varepsilon < 15\ k_{\mathrm{B}}T$ (so that $\mu < 20$ ms). In addition, with $10\ k_{\mathrm{B}}T < \varepsilon < 15\ k_{\mathrm{B}}T$ the average energy consumption per ms turns out to be less than 1 $k_{\mathrm{B}}T$/ms for the choices $u = 1 \div 4\ k_{\mathrm{B}}T$, which is in agreement with the biological evidence.

Next, the role played by the positions of the potential wells has been analyzed. As Figure 7 shows, whatever γ the mean energy consumption per ms, $\mathcal{H}$, is a decreasing function of the load's magnitude $|F_e|$. Furthermore, $\mathcal{H}$ decreases as the position L_A of the first potential well moves away from $L/2$, leftbound. As for the role of δ, $\mathcal{H}$ is seen to exhibit a minimum that depends on F_e. This minimum (see Figure 8) is located at $(-L/10, 0)$ whenever F_e takes biologically admissible values. In agreement with the above determined parameters ranges, let $u = 5$, $\varepsilon = 11.7$, $\delta = -0.15$ and $\gamma = -0.9$. (This implies that the first well is at 1.85 nm while the second well is at 4.45 nm). For such set of values the mean exit time μ and forward step probability p are shown in Figure 9, whereas Figure 10 shows the mean energy consumption per ms, $\mathcal{H}$, and the motor efficiency η. Note that

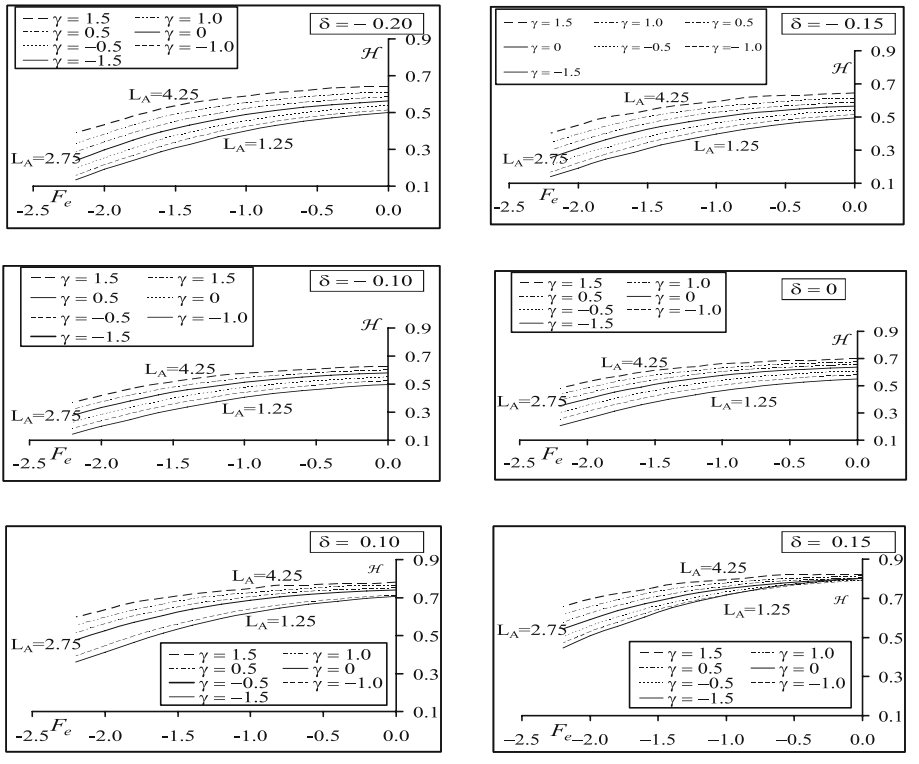


Fig. 7. The energy consumption per ms $\mathcal{H}$ is plotted as function of the load F_e for some values of γ. (Recall that the first potential well is at $L_A = L/2 + \gamma$) and that the second well is at $L_A + L + \delta$). In each box the chosen values of δ is indicated.

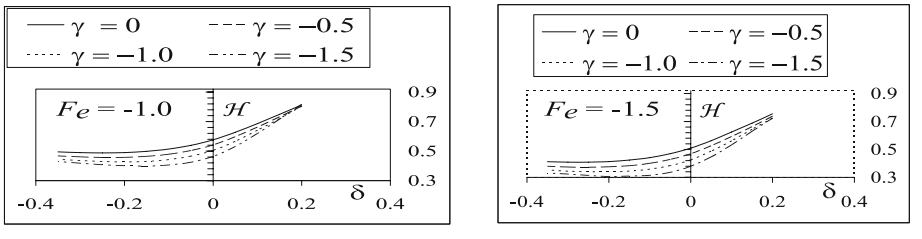


Fig. 8. For the two specified loads, the behavior of $\mathcal{H}$ as a function of δ is shown for various values of γ. Each curve has a minimum around the left of the origin. The location of such minimum moves further away from the origin as the magnitude of F_e increases.

for load values between -2 pN and -1 pN, $\mathcal{H}$ is less than 0.5 $k_{\mathrm{B}}T$/ms, while the motor efficiency stays between 30% and 50%.

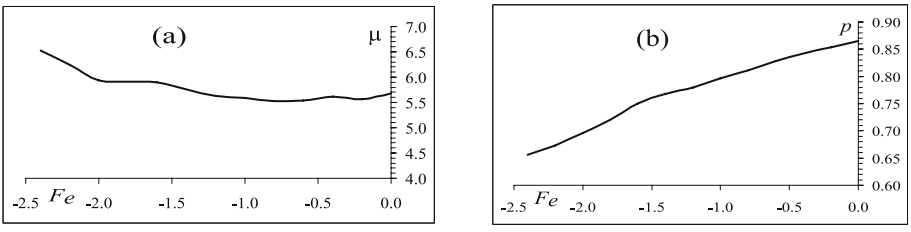


Fig. 9. Plots of mean first-exit time (ms) (a) and of forward step probability (b) versus F_e (pN). Here $a_1 = a_2 = 1.5 \cdot 10^{-1}$/ms, $L = 5.5$ nm, $\rho = 9 \cdot 10^{-5}$ pN·ms/nm, $T = 293$ °K, $U_0 = 32\ k_{\mathrm{B}}T$, $u = 5\ k_{\mathrm{B}}T$, $\varepsilon = 11.7\ k_{\mathrm{B}}T$, $\delta = -0.15$ nm and $\gamma = -0.9$ nm.

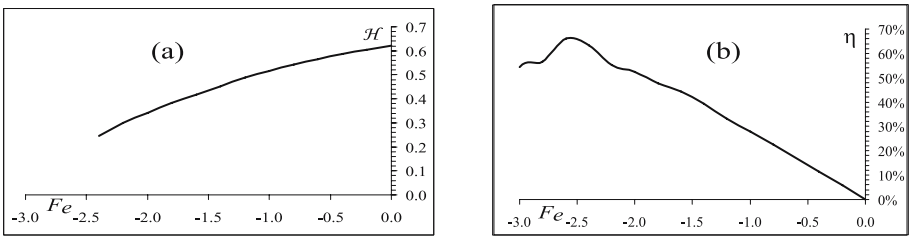


Fig. 10. Plots of mean energy consumption per ms (a) and of the motor efficiency (b) versus F_e (pN). Here all parameters are as in Figure 9.

References

1. Astumian, R.D., Bier, M.: Fluctuation Driven Ratchets: Molecular Motors. Physical Review Letters 72, 1766–1769 (1994)
2. Buonocore, A., Ricciardi, L.M.: Exploiting Thermal Noise for an Efficient Actomyosin Sliding Mechanism. Mathematical Biosciences 182, 135–149 (2003)
3. Buonocore, A., Caputo, L., Ishii, Y., Pirozzi, E., Yanagida, T., Ricciardi, L.M.: On Myosin II dynamics in the presence of external loads. Biosystems 81, 161–165 (2005)
4. Buonocore, A., Caputo, L., Pirozzi, E., Ricciardi, L.M.: On Myosin II dynamics: from Pulsating Ratchet to a Washboard Potential. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 426–435. Springer, Heidelberg (2005)
5. Buonocore, A., Caputo, L., Pirozzi, E., Ricciardi, L.M.: On a pulsating Brownian motor and its characterization. Mathematical Biosciences 207, 387–401 (2007)
6. Chauwin, J.F., Ajdari, A., Prost, J.: Force-Free Motion in Asymmetric Structures: a Mechanism without Diffusive Steps. Europhysics Letters 27, 421–426 (1994)
7. Kitamura, K., Tokunaga, M., Iwane, A.H., Yanagida, T.: A single myosin head moves along an actin filament with regular steps of 5.3 nanometres. Nature 397, 129–134 (1999)
8. Makhnovskii Yu, A., Rozenbaum, V.M., Yang, D.-Y., Lin, S.H., Tsong, T.Y.: Flashing ratchet model with high efficiency. Phys. Rev. E 69, 1–7 (2004)

9. Reimann, P.: Brownian motors: noisy transport far from equilibrium. Phys. Rep. 361, 57–265 (2002)
10. Sekimoto, K.: Kinetic Characterization of Heat Bath and the Energetics of Thermal Ratchet Models. Journal of the Physical Society of Japan 66, 1234–1237 (1997)
11. Wang, H., Oster, G.: Ratchets, power strokes, and molecular motors. Applied Physics A 75, 315–323 (2002)

Random Motion with Gamma-Distributed Alternating Velocities in Biological Modeling*

Antonio Di Crescenzo and Barbara Martinucci

Dipartimento di Matematica e Informatica, Università di Salerno
Via Ponte don Melillo, I-84084 Fisciano (SA), Italy
{adicrescenzo,bmartinucci}@unisa.it

Abstract. Motivated by applications in mathematical biology concerning randomly alternating motion of micro-organisms, we analyze a generalized integrated telegraph process. The random times between consecutive velocity reversals are gamma-distributed, and perform an alternating renewal process. We obtain the probability law and the mean of the process.

1 Introduction

The telegraph random process describes the motion of a particle on the real line, traveling at constant speed, whose direction is reversed at the arrival epochs of a Poisson process. After some initial works, such as [8], [11] and [17], numerous efforts have been made by numerous authors and through different methods to analyze this process. Various results on the telegraph process, including the first-passage-time density and the distribution of motion in the presence of reflecting and absorbing barriers have been obtained in [6], [7] and [19]. A wide and comprehensive review devoted to this process has recently been offered by Weiss [22], who also emphasized its relations with some physical problems.

In various applications in biomathematics the telegraph process arises as a stochastic model for systems driven by dichotomous noise (see [2], for instance). Two stochastic processes modeling the major modes of dispersal of cells or organisms in nature are introduced in [20]; under certain assumptions, the motion consisting of sequences of runs separated by reorientations with new velocities is shown to be governed by the telegraph equation. Moreover, the discrete analog of the telegraph process, i.e. the correlated random walk, is usually used as a model of the swarming behavior of myxobacteria (see [5], [9], [13] and [15]). Processes governed by hyperbolic equations are also used to describe movement and interaction of animals [16] and chemotaxis [10]. Moreover, the integrated telegraph process has been also used to model wear processes [3] and to describe the dynamics of the price of risky assets [4].

Many authors proposed suitable generalizations of the telegraph process, such as the 1-dimensional cases with three cyclical velocities [18], or with n values of

* Work performed under partial support by MIUR (PRIN 2005), G.N.C.S.-INdAM and Regione Campania.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 163–170, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the velocity [12], or with random velocities [21]. See also the paper by Lachal [14], where the cyclic random motion in $\mathbb{R}^d$ with n directions is studied.

A generalized integrated telegraph process whose random times separating consecutive velocity reversals have a general distribution and perform an alternating renewal process has been studied in [1] and [23]. Along the line of such articles, in this paper we study a stochastic model for particles motion on the real line with two alternating velocities c and $-v$. The random times between consecutive reversals of direction perform an alternating renewal process and are gamma distributed, which extends the Erlang-distribution case treated in [1].

In Section 2 we introduce the stochastic process $\{(X_t, V_t); t \geq 0\}$, with X_t and V_t denoting respectively position and velocity of the particle at time t. In Section 3 we obtain a series-form of the random motion probability law for gamma-distributed random inter-renewal times, whereas the mean value of X_t conditional on initial velocity is finally obtained in Section 4.

2 The Random Motion

We consider a random motion on $\mathbb{R}$ with two alternating velocities c and $-v$, with $c, v > 0$. The direction of motion is forward or backward when the velocity is c or $-v$, respectively. Velocities change according to the alternating counting process $\{N_t; t \geq 0\}$ characterized by renewal times $T_1, T_2, \ldots$, so that T_n is the n-th random instant in which the motion changes velocity. Hence,

$$N_0 = 0, \qquad N_t = \sum_{n=1}^{\infty} \mathbf{1}_{\{T_n \leqslant t\}}, \quad t > 0.$$

Let $\{(X_t, V_t); t \geq 0\}$ be a stochastic process on $\mathbb{R} \times \{-v, c\}$, where X_t and V_t give respectively position and velocity of the motion at time t. Assuming that $X_0 = 0$ and $v_0 \in \{-v, c\}$, for $t > 0$ we have:

$$X_t = \int_0^t V_s \, \mathrm{d}s, \qquad V_t = \frac{1}{2}(c - v) + \mathrm{sgn}(V_0) \, \frac{1}{2}(c + v) \, (-1)^{N_t}. \tag{1}$$

Denoting by U_k (D_k) the duration of the k-th time interval during which the motion goes forward (backward), we assume that $\{U_k; k = 1, 2, \ldots\}$ and $\{D_k; k = 1, 2, \ldots\}$ are mutually independent sequences of independent copies of non-negative and absolutely continuous random variables U and D.

If the motion does not change velocity in $[0, t]$, then $X_t = V_0 \, t$. Otherwise, if there is at least one velocity change in $[0, t]$, then $-vt < X_t < ct$ w.p. 1. Hence, the conditional law of $\{(X_t, V_t); t \geq 0\}$ is characterized by a discrete component

$$\mathbf{P}\{X_t = yt, V_t = y \,|\, X_0 = 0, V_0 = y\},$$

and by an absolutely continuous component

$$p(x, t \,|\, y) = f(x, t \,|\, y) + b(x, t \,|\, y), \tag{2}$$

where

$$f(x,t\,|\,y)=\frac{\partial}{\partial x}\mathbf{P}\{X_t\le x, V_t=c\,|\,X_0=0,V_0=y\},$$
$$b(x,t\,|\,y)=\frac{\partial}{\partial x}\mathbf{P}\{X_t\le x, V_t=-v\,|\,X_0=0,V_0=y\},$$

with $t>0$, $-vt<x<ct$ and $y\in\{-v,c\}$.

The formal conditional law of $\{(X_t,V_t);t\ge 0\}$ has been given in Theorem 2.1 of [1] for $V_0=c$. Case $V_0=-v$ can be treated by symmetry.

Explicit results for the probability law have been obtained in Theorem 3.1 of [1] when the random times U and D separating consecutive velocity reversals have Erlang distribution. This case describes the random motion of particles subject to collisions arriving according to a Poisson process with rate λ if the motion is forward and rate μ it is backward. When the motion has initial velocity c $(-v)$, then the first $n-1$ $(r-1)$ collisions have no effect, whereas the nth (rth) collision causes a velocity reversal. In the following section we shall treat the more general case in which the random inter-renewal times are gamma distributed.

3 Gamma-Distributed Random Times

We assume that the random times U and D are gamma distributed with parameters (λ,α) and (μ,β), respectively, where $\lambda,\mu>0$ and $\alpha,\beta>0$. Hereafter we obtain the probability law of $\{(X_t,V_t);t\ge 0\}$ for this case.

Theorem 1. *If U and D are gamma-distributed with parameters (λ,α) and (μ,β), respectively, for $t>0$ it is*

$$\mathbf{P}\{X_t=ct, V_t=c\,|\,X_0=0,V_0=c\}=\frac{\Gamma(\alpha,\lambda t)}{\Gamma(\alpha)}, \tag{3}$$

and, for $-vt<x<ct$,

$$f(x,t\,|\,c)=\frac{1}{c+v}\left\{e^{-\mu\overline{x}}\sum_{k=1}^{+\infty}\frac{\mu^{k\beta}(\overline{x})^{k\beta-1}}{\Gamma(k\beta)}\Big[P(k\alpha,\lambda x^*)-P(k\alpha+\alpha,\lambda x^*)\Big]\right\}, \tag{4}$$

$$b(x,t\,|\,c)=\frac{1}{c+v}\left\{\frac{\lambda^\alpha e^{-\lambda x^*}(x^*)^{\alpha-1}}{\Gamma(\alpha)}\frac{\Gamma(\beta,\mu\overline{x})}{\Gamma(\beta)}\right.$$
$$\left.+e^{-\lambda x^*}\sum_{k=1}^{+\infty}\frac{\lambda^{(k+1)\alpha}(x^*)^{(k+1)\alpha-1}}{\Gamma((k+1)\alpha)}\Big[P(k\beta,\mu\overline{x})-P(k\beta+\beta,\mu\overline{x})\Big]\right\}, \tag{5}$$

where

$$\overline{x}=\overline{x}(x,t)=\frac{ct-x}{c+v},\qquad x^*=x^*(x,t)=\frac{vt+x}{c+v},$$

and

$$\Gamma(a,u)=\int_u^\infty t^{a-1}e^{-t}\mathrm{d}t,\qquad P(a,u)=\frac{1}{\Gamma(a)}\int_0^u t^{a-1}e^{-t}\mathrm{d}t,\qquad a>0. \tag{6}$$

Proof. Making use of (2.4) of [1] and noting that for $k \geq 1$ the pdfs of $U^{(k)} = U_1 + \ldots + U_k$ e $D^{(k)} = D_1 + \ldots + D_k$ are given by

$$f_U^{(k)}(x) = \frac{\lambda^{k\alpha} x^{k\alpha-1} e^{-\lambda x}}{\Gamma(k\alpha)}, \qquad f_D^{(k)}(x) = \frac{\mu^{k\beta} x^{k\beta-1} e^{-\mu x}}{\Gamma(k\beta)}, \qquad x > 0, \tag{7}$$

we have

$$f(x,t\,|\,c) = \frac{1}{c+v}\, e^{-\mu\overline{x}} e^{\lambda\overline{x}} \sum_{k=1}^{+\infty} \frac{\mu^{k\beta}(\overline{x})^{k\beta-1}}{\Gamma(k\beta)} \frac{\lambda^{k\alpha}}{\Gamma(k\alpha)\Gamma(\alpha)}\, \mathcal{I}_k, \tag{8}$$

where

$$\mathcal{I}_k := \int_{\overline{x}}^{t} e^{-\lambda s} (s-\overline{x})^{k\alpha-1} \Gamma(\alpha, \lambda(t-s))\, \mathrm{d}s, \qquad k \geq 1. \tag{9}$$

Noting that, due to (6), $\Gamma(a,u) = \Gamma(a)\left[1 - P(a,u)\right]$ we obtain

$$\mathcal{I}_k = \mathcal{I}_{1,k} - \mathcal{I}_{2,k}, \tag{10}$$

where, for $k \geq 1$

$$\mathcal{I}_{1,k} := \Gamma(\alpha) \int_{\overline{x}}^{t} e^{-\lambda s} (s-\overline{x})^{k\alpha-1}\, \mathrm{d}s = \Gamma(k\alpha)\, \Gamma(\alpha)\, e^{-\lambda\overline{x}} \lambda^{-k\alpha} P(k\alpha, \lambda x^*), \tag{11}$$

$$\begin{aligned} \mathcal{I}_{2,k} &:= \Gamma(\alpha) \int_{\overline{x}}^{t} e^{-\lambda s} (s-\overline{x})^{k\alpha-1}\, P(\alpha, \lambda(t-s))\, \mathrm{d}s \\ &= e^{-\lambda\overline{x}} \Gamma(\alpha) \int_0^{x^*} e^{-\lambda y} y^{k\alpha-1}\, P(\alpha, \lambda(x^*-y))\, \mathrm{d}y = e^{-\lambda\overline{x}} \lambda^{-k\alpha}\, G(\lambda x^*), \end{aligned} \tag{12}$$

with

$$G(\lambda x^*) := \int_0^{\lambda x^*} e^{-u} u^{k\alpha-1} \left(\int_0^{\lambda x^* - u} e^{-\tau} \tau^{\alpha-1}\, \mathrm{d}\tau \right) \mathrm{d}u. \tag{13}$$

Making use of the Laplace transform of (13) it follows

$$\mathcal{L}_z\{G(\lambda x^*)\} = \mathcal{L}_z\left\{ e^{-\lambda x^*} (\lambda x^*)^{k\alpha-1} \right\} \mathcal{L}_z\left\{ \int_0^{\lambda x^*} e^{-\tau} \tau^{\alpha-1} \mathrm{d}\tau \right\} = \frac{\Gamma(k\alpha)\, \Gamma(\alpha)}{z(z+1)^{k\alpha+\alpha}}.$$

Hence, from identity

$$\mathcal{L}_z\left\{P(k\alpha+\alpha, \lambda x^*)\right\} = \mathcal{L}_z\left\{ \int_0^{\lambda x^*} \frac{u^{k\alpha+\alpha-1} e^{-u}}{\Gamma(k\alpha+\alpha)}\, \mathrm{d}u \right\} = \frac{1}{z(z+1)^{k\alpha+\alpha}},$$

we have

$$G(\lambda x^*) = \Gamma(k\alpha)\, \Gamma(\alpha)\, P(k\alpha+\alpha, \lambda x^*).$$

Eqs. (10)÷(13) thus give

$$\mathcal{I}_k = \Gamma(k\alpha)\, \Gamma(\alpha)\, e^{-\lambda\overline{x}} \lambda^{-k\alpha} \left[P(k\alpha, \lambda x^*) - P(k\alpha+\alpha, \lambda x^*) \right]. \tag{14}$$

Eq. (4) then follows from (8) and (13). In order to obtain $b(x,t\,|\,c)$, we recall (2.5) of [1] and make use of (7) to obtain

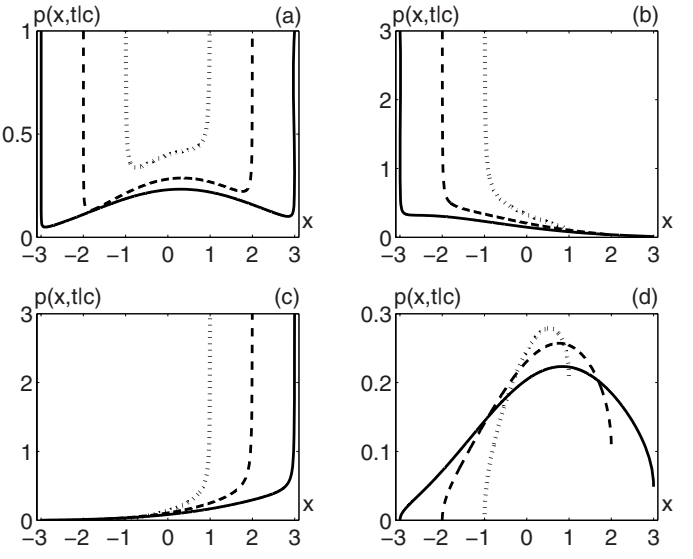


Fig. 1. Density (2) for $c = v = 1$, $\lambda = \mu = 1$, and $t = 1$ (dotted line), $t = 2$ (dashed line), and $t = 3$ (solid line), with (a) $\alpha = \beta = 0.5$, (b) $\alpha = 0.5$ and $\beta = 1.5$, (c) $\alpha = 1.5$ and $\beta = 0.5$, (d) $\alpha = \beta = 1.5$

$$b(x,t\,|\,c) = \frac{1}{c+v}\left\{\frac{\lambda^\alpha e^{-\lambda x^*}(x^*)^{\alpha-1}}{\Gamma(\alpha)}\frac{\Gamma(\beta,\mu\overline{x})}{\Gamma(\beta)}\right.$$
$$+\ e^{-\lambda x^*}e^{\mu x^*}\sum_{k=1}^{+\infty}\frac{\lambda^{(k+1)\alpha}(x^*)^{(k+1)\alpha-1}}{\Gamma((k+1)\alpha)}$$
$$\left.\times\ \frac{\mu^{k\beta}}{\Gamma(\beta)\Gamma(k\beta)}\int_{x^*}^{t} e^{-\mu s}(s-x^*)^{k\beta-1}\Gamma(\beta,\mu(t-s))\mathrm{d}s\right\}. \qquad (15)$$

Due to (9), the integral in (15) can be calculated from (14) by interchanging x^*, β, μ with $\overline{x}$, α, λ, respectively. Eq. (5) then follows after some calculations. □

Figure 1 shows density $p(x,t\,|\,c)$ as x varies in $(-vt, ct)$ for various choices of t, α and β. Hereafter we analyze the obtain the limits of densities $f(x,t\,|\,c)$ and $b(x,t\,|\,c)$ at the extreme points of interval $(-vt, ct)$, for any fixed t.

Proposition 1. *Under the assumptions of Theorem 1 we have*

$$\lim_{x\downarrow -vt} f(x,t|c) = 0, \quad \lim_{x\uparrow ct} f(x,t|c) = \begin{cases} +\infty, & 0<\beta<1 \\ \dfrac{\mu}{c+v}\big[P(\alpha,\lambda t) - P(2\alpha,\lambda t)\big], & \beta = 1 \\ 0, & \beta > 1, \end{cases}$$

$$\lim_{x\uparrow ct} b(x,t|c) = \frac{\lambda^\alpha \mathrm{e}^{-\lambda t}t^{\alpha-1}}{(c+v)\Gamma(\alpha)}, \quad \lim_{x\downarrow -vt} b(x,t|c) = \begin{cases} +\infty, & 0<\alpha<1 \\ \dfrac{\lambda\,\Gamma(\beta,\mu t)}{(c+v)\Gamma(\alpha)\Gamma(\beta)}, & \alpha = 1 \\ 0, & \alpha > 1. \end{cases}$$

From Proposition 1 we note that if $\alpha < 1$ ($\beta < 1$), i.e. the gamma inter-renewal density has a decreasing hazard rate, then the backward (forward) density is divergent when x approaches $-vt$ (ct). This is very different from the behavior exhibited in the case of Erlang-distributed inter-renewals (see Corollary 3.1 of [1]), when the limits are finite.

4 Mean Value

In this Section we obtain the mean value of X_t when random times U and D are identically gamma distributed.

Theorem 2. *Let U and D have gamma distribution with parameters (λ, α). For any fixed $t \in (0, +\infty)$, we have*

$$E\big[X_t \,\big|\, V_0\big] = V_0\, t + \frac{c+v}{\lambda}\,\mathrm{sgn}(V_0) \sum_{k=1}^{+\infty} (-1)^k \Big[\lambda t\, P(k\alpha, \lambda t) - k\alpha P(k\alpha + 1, \lambda t)\Big]. \tag{16}$$

Proof. Due to Eqs. (1) and recalling that $\mathbf{P}(T_k \le s) = P(k\alpha, \lambda s)$, $s \ge 0$, it is

$$\begin{aligned} E\big[X_t \,\big|\, V_0\big] &= \frac{1}{2}(c-v)t + \frac{1}{2}(c+v)\,\mathrm{sgn}(V_0) \int_0^t E\left[(-1)^{N_s}\right] \mathrm{d}s \qquad (17)\\ &= \frac{1}{2}(c-v)t + \frac{1}{2}(c+v)\,\mathrm{sgn}(V_0) \int_0^t \left\{1 + 2\sum_{k=1}^{+\infty} (-1)^k\, P(k\alpha, \lambda s)\right\} \mathrm{d}s \\ &= V_0\, t + (c+v)\,\mathrm{sgn}(V_0) \sum_{k=1}^{+\infty} (-1)^k \int_0^t P(k\alpha, \lambda s)\, \mathrm{d}s. \end{aligned}$$

(Note that the above series is uniformly convergent.) Moreover, recalling (6) it is not hard to see that

$$\int_0^t P(k\alpha, \lambda s)\, \mathrm{d}s = t\, P(k\alpha, \lambda t) - \frac{k\alpha}{\lambda}\, P(k\alpha + 1, \lambda t).$$

Eq. (16) then immediately follows. □

The graphs given in Figure 2 show the mean value of X_t conditional on $V_0 = c$ for some choice of the involved parameters. We note that, being $P(\alpha, t) \sim t^{\alpha-1}/\Gamma(\alpha)$ as $t \to 0$, under the assumptions of Theorem 2 from (16) we have

$$E\big[X_t \,\big|\, V_0\big] \sim V_0\, t \qquad \text{as } t \to 0.$$

We remark that when $\alpha = n$ is integer, i.e. the random times U and D are Erlang-distributed with parameters (λ, n), then $E\big[X_t \,\big|\, V_0\big]$ can be computed making use of (17) and noting that

$$E\left[(-1)^{N_s}\right] = 1 - 2\mathrm{e}^{-\lambda s} \sum_{k=0}^{+\infty} \sum_{j=2nk+n}^{2nk+2n-1} \frac{(\lambda s)^j}{j!}.$$

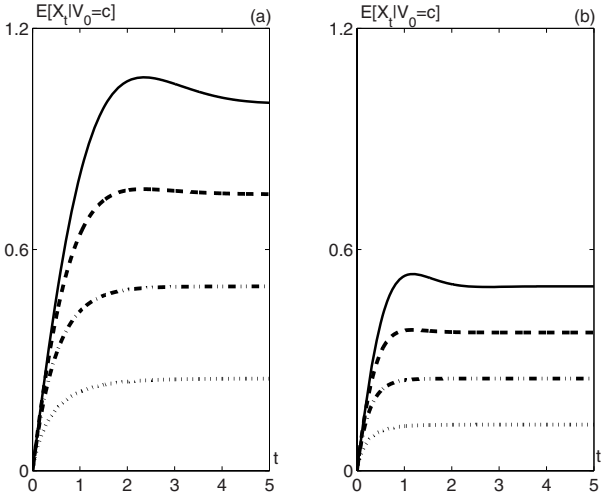


Fig. 2. Mean value $E[X_t \,|\, V_0 = c]$, for $c = v = 1$ and $\alpha = 0.5$ (dotted line), $\alpha = 1$ (dash-dot line), $\alpha = 1.5$ (dash line), $\alpha = 2$ (solid line), with (a) $\lambda = 1$ and (b) $\lambda = 2$

For instance, in this case the following expressions hold for $t > 0$:

n	$E\big[X_t \,\big|\, V_0\big]$
1	$\dfrac{(c-v)t}{2} + \dfrac{(c+v)}{4\lambda}\,\mathrm{sgn}(V_0)\,[1 - \mathrm{e}^{-2\lambda t}]$
2	$\dfrac{(c-v)t}{2} + \dfrac{(c+v)}{2\lambda}\,\mathrm{sgn}(V_0)\,[1 - \mathrm{e}^{-\lambda t}\cos(\lambda t)]$
3	$\dfrac{(c-v)t}{2} + \dfrac{(c+v)}{2\lambda}\,\mathrm{sgn}(V_0)\left\{\dfrac{1-\mathrm{e}^{-2\lambda t}}{6} + \dfrac{4}{3}[1 - \mathrm{e}^{-\frac{\lambda t}{2}}\cos(\dfrac{\sqrt{3}}{2}\lambda t)]\right\}$
4	$\dfrac{(c-v)t}{2} + \dfrac{(c+v)}{2\lambda}\,\mathrm{sgn}(V_0)\left\{\big[1 - (1 + \dfrac{\sqrt{2}}{2})\mathrm{e}^{-\lambda t(1-\frac{\sqrt{2}}{2})}\cos(\dfrac{\sqrt{2}}{2}\lambda t)\big]\right.$ $\left. + \big[1 - (1 - \dfrac{\sqrt{2}}{2})\mathrm{e}^{-\lambda t(1+\frac{\sqrt{2}}{2})}\cos(\dfrac{\sqrt{2}}{2}\lambda t)\big]\right\}$

References

1. Di Crescenzo, A.: On random motions with velocities alternating at Erlang-distributed random times. Advances in Applied Probability 33, 690–701 (2001)
2. Di Crescenzo, A., Martinucci, B.: On the effect of random alternating perturbations on hazard rates. Scientiae Mathematicae Japonicae 64, 381–394 (2006)
3. Di Crescenzo, A., Pellerey, F.: Stochastic comparison of wear processes characterized by random linear wear rates. In: Mikulin, M., Limnios, N. (eds.) 2nd International Conference on Mathematical Methods in Reliability, Abstracts Book, Bordeaux, vol. 1, pp. 339–342 (2001)

4. Di Crescenzo, A., Pellerey, F.: On Prices' evolutions based on geometric telegrapher's process. Applied Stochastic Models in Business and Industry 18(2), 171–184 (2002)
5. Erdmann, U., Ebeling, W., Schimansky-Geier, L., Ordemann, A., Moss, F.: Active Brownian particle and random walk theories of the motions of zooplankton: Application to experiments with swarms of Daphnia. arXiv:q-bio.PE/0404018
6. Foong, S.K.: First-passage time, maximum displacement, and Kac's solution of the telegrapher equation. Physical Review A 46, 707–710 (2001)
7. Foong, S.K., Kanno, S.: Properties of the telegrapher's random process with or without a trap. Stochastic Processes and their Applications 53, 147–173 (1994)
8. Goldstein, S.: On diffusion by discontinuous movements and the telegraph equation. Quarterly Journal of Mechanics and Applied Mathematics 4, 129–156 (1951)
9. Hill, N.A., Hader, D.P.: A biased random walk model for the trajectories of swimming micro-organisms. Journal of Theoretical Biology 186, 503–526 (1997)
10. Hillen, T., Stevens, A.: Hyperbolic models for chemotaxis in 1-D. Nonlinear Analysis: Real World Applications 1, 409–433 (2000)
11. Kac, M.: A stochastic model related to the telegrapher's equation. Rocky Mountain Journal of Mathematics 4, 497–509 (1974)
12. Kolesnik, A.: The equations of Markovian random evolution on the line. Journal of Applied Probability 35, 27–35 (1998)
13. Komin, N., Erdmann, U., Schimansky-Geier, L.: Random walk theory applied to Daphnia Motion. Fluctuation and Noise Letters 4(1), 151–159 (2004)
14. Lachal, A.: Cyclic random motions in $\mathbb{R}^d$-space with n directions. ESAIM Probability and Statistics 10, 277–316 (2006)
15. Lutscher, F., Stevens, A.: Emerging Patterns in a Hyperbolic Model for Locally Interacting Cell Systems. Journal of Nonlinear Science 12, 619–640 (2002)
16. Lutscher, F.: Modeling alignment and movement of animals and cells. Journal of Mathematical Biology 45, 234–260 (2002)
17. Orsingher, E.: Probability law, flow function, maximum distribution of wave-governed random motions and their connections with Kirchoff's laws. Stochastic Processes and their Applications 34, 49–66 (1990)
18. Orsingher, E.: Random motions governed by third-order equations. Advances in Applied Probability 22, 915–928 (1990)
19. Orsingher, E.: Motions with reflecting and absorbing barriers driven by the telegraph equation. Random Operators and Stochastic Equations 3, 9–21 (1995)
20. Othmer, H.G., Dunbar, S.R., Alt, W.: Models of dispersal in biological systems. Journal of Mathematical Biology 26(3), 263–298 (1988)
21. Stadje, W., Zacks, S.: Telegraph processes with random velocities. Journal of Applied Probability 41, 665–678 (2004)
22. Weiss, G.H.: Some applications of persistent random walks and the telegrapher's equation. Physica A 311, 381–410 (2002)
23. Zacks, S.: Generalized integrated telegrapher process and the distribution of related stopping times. Journal of Applied Probability 41, 497–507 (2004)

A Prey-Predator Model for Immune Response and Drug Resistance in Tumor Growth*

G. Albano[1], V. Giorno[1], and C. Saturnino[2]

[1] Dipartimento di Matematica e Informatica, Università di Salerno, Via Ponte don Melillo, Fisciano (SA), Italy
{pialbano,giorno}@unisa.it
[2] Dip. di Scienze Farmaceutiche , Università di Salerno, Via Ponte don Melillo, Fisciano (SA), Italy
saturnino@unisa.it

Abstract. In this paper the interaction between cancer and immunological system is analysed. The model is a modification of a one proposed by de Vladar and Gonzales. For our model the stability is analysed. Furthermore, the effect of a therapy schedule with fixed concentration is investigated. Mathematically, the effect of therapy is viewed as a moderating term both for tumor cell growth rate and for T-cells death rate. In conclusion, a systematic computational analysis is associated to the obtained theoretical results.

Keywords: Prey-predator model; Gompertz growth; immune surveilance.

1 Introduction

Many mathematical models have been proposed in literature to describe the evolution of tumors (see, for instance, [2], [5], [8]). Some of these models are based on the Gompertz law because this one approximates experimental observations of cancer growth. Furthermore, to take into account environmental fluctuations, in [1] a stochastic approach generalizing Gompertz growth has been performed.

The aim of this paper is to describe the interaction between the cancer and the immunological system. In this direction, in [2] de Vladar and Gonzales proposed a model that includes the principal characteristics of the T-cell-mediated reaction against cancer. This model is described by two ordinary differential equations:

$$\begin{aligned} \dot{x} &= \alpha\, x - \xi\, x\, \ln x - \gamma\, x\, y \\ \dot{y} &= \mu_I\, (x - \beta x^2)\, y - \delta\, y + k \end{aligned} \tag{1}$$

where x and y denote the size of tumor and of the immunocompetent cells respectively, and the parameters α and ξ are the growth and decay rates of tumor cells; γ is the rate of elimination of cancer cells by the activity of T-cells; μ_I

* Work performed under partial support by MIUR (PRIN 2005) G.N.C.S-INdAM.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 171–178, 2007.
© Springer-Verlag Berlin Heidelberg 2007

represents the production of T-cells due to the stimulation of cancer cells; β^{-1} denotes the saturation density above which the immunological system is suppressed; δ is the natural death rate of T-cells and k is the constant rate of influx of T-cells from primary organs. We choose the description of de Vladar-Gonzales model since it comprises the relevant features of cancer growth under immunological activity. On the other hand, in de Vladar-Gonzales model the growth rate of immunocompetent cells depends on tumor size; when x reaches $1/\beta$ this rate, it becomes negative implying that tumor cells are able to kill the cells activated from immunological system. According to de Pillis (cf. [7]), we believe that immunocompetent cells eventually become inactivated after some number of interactions with tumor cells, but it doesn't exist any biological mechanism which allows to the tumor cells to kill T-cells. Indeed, the well known phenomenon of leukopenia, i.e. a decrease in the number of leukocytes in the blood, is associated to chemotherapy or to radiation therapy. In the second section of this work we will review the main characteristics of the model by de Vladar-Gonzales to motivate and justify our modification. In the third section we will describe the features of our model and, in the Section 4, we will study the behaviour of the system in the presence of typical treatment schedules including drug resistance. Finally, in the last section, the possibility to include "random environment" in the model is investigated.

2 de Vladar-Gonzales Model

System (1) satisfies an heuristic but desiderable property in tumor models: the point $P_0 \equiv (0, k/\delta)$ is an unstable fixed point for all choices of the parameters. So the model predicts that the tumor grows aggressively near the zero, therefore there is no possible condition that allows self-regression. This also implies that treatments which leave microscopic amounts of cancer cells (subclinical unobservable cancer) will permit a new tumor growth. We note that the quantity k/δ is the basal density of immune-effectors in the absence of tumor cells.

This model is also able to describe the different types of tumor behaviour observed clinically: first, the description of dormant tumors which do not grow more than a finite size; second, the description of malignant tumors which clinically can be considered of unlimited size.

2.1 de Vladar-Gonzales Model Under Therapy

de Vladar-Gonzales consider that the effect of a therapy consists in the elimination of a proportion of cells. Therapy affects both tumor cells and T-cells in a proportion k_C and k_I. Mathematically the equation (1) becomes:

$$\begin{aligned}\dot{x} &= \alpha\, x - \xi\, x\, \ln x - \gamma\, x\, y - k_C\, C(t)\\ \dot{y} &= \mu_I\, (x - \beta x^2)\, y - \delta\, y + k - k_I\, C(t),\end{aligned}$$

so, in the case of a therapy with fixed dosage C_0 (cf. [6]) the effect is a parametric perturbation on the system (1), indeed

$$\alpha \to \alpha - k_C\, C_0,$$
$$\delta \to \delta + k_I C_0.$$

Then, in accordance to clinical experience, the model predicts that treatment applications induce shrinking of the tumor to microscopical tumors and not to self-regression.

We point out that the model (1) neglects one of the principal causes of the failure of chemotherapeutic treatment, i.e. the resistance development of cancer. So another description qualitatively compatible with the observations is necessary.

Since we are interested in therapeutic treatment, we will consider induced resistance, i.e. that comes from the use of chemotherapy.

In the following section, we will include a Richer's function in de Vladar-Gonzales model to improve the description of the interaction between the immunological activity and the cancer growth. Furthermore, in the mathematical model, drug resistance will be obtained by a progressive decrease of drug intensity.

3 The Model

Our model is described by the following system:

$$\begin{aligned} \dot{x} &= \alpha\, x - \xi\, x\, \ln x - \gamma\, x\, y \\ \dot{y} &= \mu_I\, x\, e^{-\lambda\, x} - \delta\, y + k. \end{aligned} \tag{2}$$

In this modified scheme the meaning of the parameter λ is analogous with respect to β in the previous model. We are assuming that the proliferation rate of immunocompetent cells is a concave function with respect to tumor size, so immunological system is asymptotically vanished from cancer cells (see [7]). As in de Vladar-Gonzales's model, our description presents an unstable fixed point at $P_0 \equiv (0, \frac{k}{\delta})$ so for tumor of small size, the acceleration of the growth is such that any interaction from the immunological system is unattainable. Moreover, if $x \neq 0$, further equilibrium points for the system (3) can be obtained as points of intersection between the two nullclines. They have the following equations:

$$\begin{aligned} y_C &= -\frac{\alpha}{\gamma} - \frac{\xi}{\gamma} \log x; \\ y_I &= -\frac{k}{\mu_I\, x\, e^{-\lambda\, x} - \delta}. \end{aligned} \tag{3}$$

We observe that y_I represents the immunocompetent cells size, so $y_I > 0 \quad \forall\, x$. Hence, setting $A = \mu_I\, x\, e^{-\lambda\, x} - \delta$, from (3)it is easy to see that $A < 0 \quad \forall\, x$.

To determine the stability of a fixed point, we use the method of the eigenvalues. For each equilibrium point $P^* \equiv (x^*, y^*)$ with $x^* \neq 0$, the jacobian matrix associated to the system (3) is:

$$\mathbf{J} = \begin{pmatrix} -\xi & \dfrac{\xi}{y'_C} \\ -A\, y'_I & A \end{pmatrix}$$

where y'_C and y'_I are the derivatives of the nullclines (3) with respect to tumor size x. So the eigenvalues are:

$$\lambda_{\pm} = \frac{A - \mu_C}{2} \pm \frac{1}{2}\sqrt{(A-\mu_C)^2 + 4\,\mu_C\,A\left(1 - \frac{y'_I}{y'_C}\right)}. \tag{4}$$

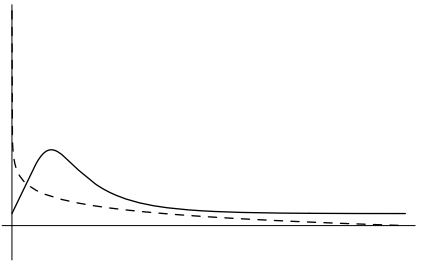


Fig. 1. Phase space in the presence of one fixed point corresponding to a dormant tumor: full line represents y_I and dashed line is y_C

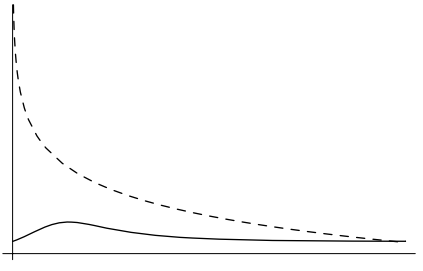


Fig. 2. Phase space in the presence of one fixed point corresponding to a malignant tumor: full line represents y_I and dashed line is y_C

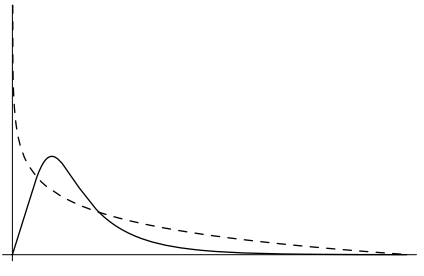


Fig. 3. Phase space in the presence of three fixed points: full line represents y_I and dashed line is y_C

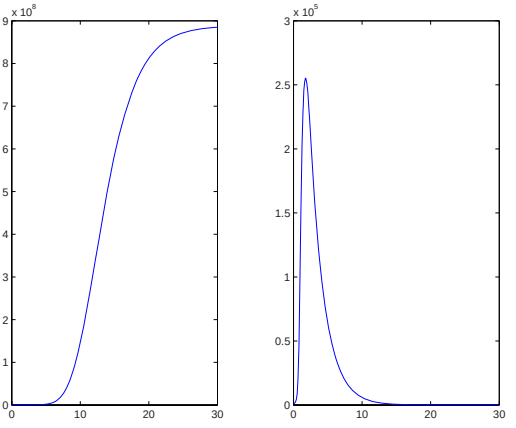


Fig. 4. Evolution of parathyroid tumor cells (left) and immunocompetent cells (right) are plotted with $x_0 = y_0 = 1000$; $\mu_I = 0.023\, years^{-1}$, $\lambda = 10^{-4}\, years^{-1}$, $\delta = 0.47\, years^{-1}$, $k = 1.07$, $\gamma = 10^{-5}\, years^{-1}$

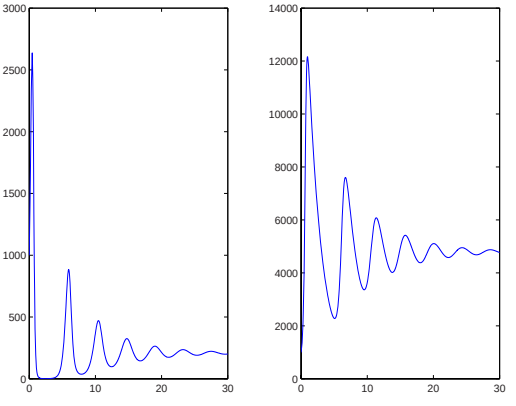


Fig. 5. Evolution of parathyroid tumor cells (left) and immunocompetent cells (right) are plotted as in Figure 4 with $\gamma = 10^{-5}\, years^{-1}$

Imposing that both eigenvalues are negative we find the condition for stable solutions. We note that $\lambda_- < 0$ since $A < 0$ for all (x^*, y^*) and

$$\lambda_+ < 0 \Leftrightarrow \frac{y_I'}{y_C'} < 1. \tag{5}$$

In Figure 1 it is shown a configuration of the phase space with one stable fixed point. Here the full line is y_I and dashed line is y_C. We note that the nullcline y_I has a maximum in $x = 1/\lambda$ so we can interpret $1/\lambda$ as the dimension supported by the organism (cf. Fig. 1). Then the fixed point P_B corresponds to a dormant tumor (if macroscopic) or simply to a microscopic equilibrium (see [3]). In

Figure 2 it is shown another possible configuration of the phase space again with one stable fixed point. Tumor size in this case is near x_∞, the carrying capacity, so biologically this point can be interpreted as a malignant and invasive tumor. The third possible configuration of the system is shown in Figure 3 in which we have three fixed points. The two fixed points previously described, namely P_B and P_M, still exist and conserve their stable nature. The point P_S, instead, is unstable (saddle point) in a suitable subset of the interval $(\frac{1}{\lambda}, x_\infty)$. In Figures 4 and 5 it's shown the evolution of the population x and y for $\gamma = 10^{-5}\ years^{-1}$ and $\gamma = 10^{-2}\ years^{-1}$ respectively, in the case of parathyroid tumor. In the first case we can see that tumor population evolves as in the absence of immunoeffectors with Gompertz law. In the second case a non trivial equilibrium point between tumor and immunological system is reached, so that our model predicts immune surveillance theory is plausible.

4 Introduction of a Therapy

The effect of a treatment schedule is evaluated on the population x and y by introducing one more term in Eq. (3). Drug resistance is obtained by a progressive decrease of aggressiveness of drug, so that a drug able to reduce tumor proliferation rate of 50% after the first administration, at the n-th ($n \geq 2$) this rate will be able to reduce it of $(50 - n\,a)\%$ ($a > 0$). So, denoted by n the number of drug administrations and by $t_i \quad (1 \leq i \leq n)$ the instants in which drug is administered, we have:

$$\dot{x} = \sum_{j=0}^{\infty} \frac{\alpha}{\gamma_j} I_{[t_j, t_{j+1})} x - \xi x \ln x - \gamma x y \tag{6}$$

$$\dot{y} = \mu_I x e^{-\lambda x} y - \delta y + k - c_0 \sum_{j=0}^{\infty} \rho_j I_{[t_j, t_{j+1})} y$$

where the parameters γ_j and ρ_j are non negative constants describing the aggressiveness of drug on the population x and y respectively, and so they are proportional to drug concentration (see [6]). The function $I_{[a,b)}(\cdot)$ denotes the indicator function of the interval $[a, b)$. We assume that the sequences $\{\gamma_j\}_{j\geq 0}$ and $\{\rho_j\}_{j\geq 0}$ are increasing until is stopped the therapeutic schedule; after $\gamma_j = 1$ and $\rho_j = 0$. In Figure 6 it is shown the evolution of tumor and immunocompetent cells in the presence of a treatment schedule with $n = 5$ and $t_j = 0.1\ years$. The parameters of the model are chosen as in Figure 4, in which immunological system is not able to control tumor. Furthermore we assume:

$$\varrho_j = \begin{cases} j+1 \text{ for } 1 \leq j < n; \\ 0 \qquad \text{otherwise;} \end{cases} \qquad \gamma_j = \begin{cases} j+1 \text{ for } 1 \leq j < n; \\ 1 \qquad \text{otherwise.} \end{cases}$$

Moreover, the final point $P_F \equiv (x(t_{n+1}), y(t_{n+1}))$ can be used to estimate the evolution of the tumor after therapy, indeed by considering P_F as initial point in Eq. (3) we obtain the dynamics of the populations x and y after the treatment

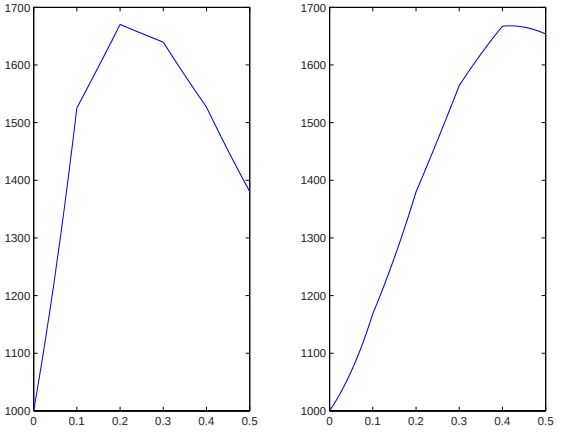


Fig. 6. Evolution of parathyroid tumor cells (left) and immunocompetent cells (right) is plotted with $x_0 = y_0 = 1000$; $\mu_I = 0.023\, years^{-1}$, $\lambda = 10^{-4}\, years^{-1}$, $\delta = 0.47\, years^{-1}$, $k = 1.07$

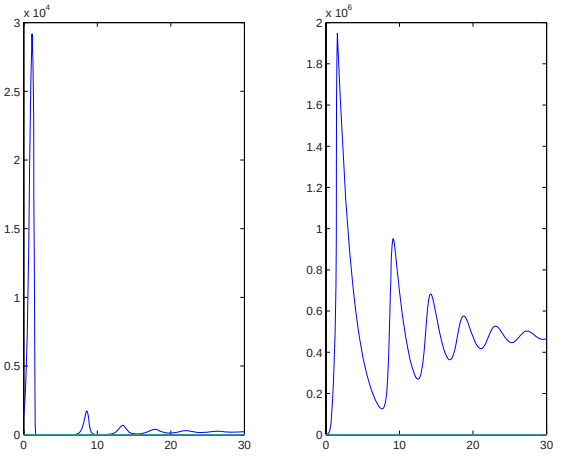


Fig. 7. Evolution of parathyroid tumor cells (left) and immunocompetent cells (right). We choose the parameters as in Figure 5

schedule. In Figure 7 it's shown the evolution of tumor and immunocompetent cells after a treatment as in Figure 6. We can see that in this case a fixed point is reached where tumor size is finite but not zero. So we conclude that a therapy with fixed concentration is not able to reduce totally tumor, but an equilibrium point between immunocompetent cells and tumor in this case is reached.

5 Further Work

In order to take into account random perturbations of the system (3), so as to model fluctuations present in experimental data, it's possible to generalize the parameters in (3) in the following way:

$$\alpha \to \alpha + \sigma_1 \, d\dot{W}_1$$
$$\delta \to \delta + \sigma_2 \, d\dot{W}_2$$

where $\dot{W}_1$ and $\dot{W}_2$ are two independent white noises and σ_1 and σ_2 are constants describing the width of random fluctuations.

So, the system (3) becomes:

$$d\,x = (\alpha\, x - \xi\, x\, \ln x - \gamma\, x\, y)\, dt + \sigma_1\, x\, dW_1(t)$$
$$d\,y = (\mu_I\, x\, e^{-\lambda\, x} - \delta\, y + k)\, dt + \sigma_2\, x\, dW_2(t).$$

In order that the system (7) has an unique solution, continuous with probability one, the coefficients have to verify the linear growth condition and the local Lipschitz condition. Such study will be subject of future work.

References

1. Albano, G., Giorno, V.: A stochastic model in tumor growth. J. Theor. Biol. 242(2), 229–236 (2006)
2. de Vladar, H.P., Gonzalez, J.A.: Dynamic response of cancer under the influence of immunological activity and therapy. J. Theor. Biol. 227(3), 335–348 (2004)
3. Dunn, G., Bruce, A., Ikeda, H., Old, L., Schreiber, R.: Cancer immunoediting: from immunosurveillance to tumor escape. Nat. Immunol. 3(11), 991–998
4. Galach, M.: Dynamics of the tumor-immune system competition-the effect of time delay. Int. J. Appl. Math. Comput. Sci. 13(3), 395–406 (2004)
5. Kozusko, F., Bajzer, Z.: Combining Gompertzian growth and cell population dynamics. Mathematical Biosciences (2003) 185, 153–167 (2004)
6. Loddo, R., Busonera, B., La Colla, P., Filona, R., Buonerba, M., Saturnino, C.: N,No (4,5-dihydro-1H-imidazol-2-yl)3-aza-1,10-decane-diamine and N,No (4,5-dihydro-1H-imidazol-2-yl)3-aza-1,10-dodecane-diamine antagonize cell proliferation as selective ligands towards the topoisomerase II, J. Pharmacy and Pharmacology (in press) (2006)
7. de Pillis, L.G., Radunskaya, A.: A mathematical model of immune response to tumor invasion. Computational Fluid and Solid Mechanics (2003)
8. Stepanova, N.: Course of the immune reaction during the development of a malignant tumor. Biophysics 24, 917–923 (1980)

On First-Passage Problems for Asymmetric One-Dimensional Diffusions

Mario Abundo

Dipartimento di Matematica, Università Tor Vergata, 00133 Roma, Italy
abundo@mat.uniroma2.it

Abstract. For $a, b > 0$, we consider a temporally homogeneous, one-dimensional diffusion process $X(t)$ defined over $I = (-b, a)$, with infinitesimal parameters depending on the sign of $X(t)$. We suppose that, when $X(t)$ reaches the position 0, it is reflected rightward to δ with probability $p > 0$ and leftward to $-\delta$ with probability $1-p$, where $\delta > 0$. It is presented a method to find approximate formulae for the mean exit time from the interval $(-b, a)$, and for the probability of exit through the right end a, generalizing the results of Lefebvre ([1]) holding, in the limit $\delta \to 0$, for asymmetric Brownian motion with drift.

Keywords: Asymmetric diffusion, Brownian motion, First-exit time.

1 Introduction

For $a, b > 0$, we consider a temporally homogeneous, one-dimensional diffusion process $X(t)$ defined over $I = (-b, a)$, which is the solution of the stochastic differential equation:

$$dX(t) = \mu(X(t))dt + \sigma(X(t))dB_t \ , \ X(0) = x \in I \tag{1}$$

where B_t is a standard Brownian motion (BM). We suppose that $\mu(X(t)) = \mu^+(X(t))$ if $X(t) > 0$, $\mu(X(t)) = \mu^-(X(t))$ if $X(t) < 0$, and $\sigma(X(t)) = \sigma^+(X(t))$ if $X(t) > 0$, $\sigma(X(t)) = \sigma^-(X(t))$ if $X(t) < 0$, where $\mu^{\pm}(x)$ and $\sigma^{\pm}(x)$ are regular enough functions satisfying the usual conditions for the existence and uniqueness of the solution of (1) with initial value in the interval $(0, a)$ and $(-b, 0)$. We can take $\mu^{\pm}(x)$ and $\sigma^{\pm}(x)$ as parametric functions of the state x. For instance, they can be polynomials in the unknown x, with coefficients $\theta_i^{\pm}$, $i = 1, \ldots, n$ and $\eta_j^{\pm}$, $j = 1, \ldots, m$, respectively. Thus, $X(t)$ turns out to be a sign-dependent diffusion. Moreover, we suppose that, when the process hits the origin, it goes right to the position $\delta > 0$ with probability $p > 0$ and left to the position $-\delta$ with probability $q = 1-p$; this means that the process is reflected at zero rightward or leftward. This kind of process is related to the so-called skew Brownian motion introduced by Itô and McKean ([2], [3]); as a generalization, we could assume that the infinitesimal coefficients $\mu(\cdot)$ and $\sigma(\cdot)$ change when the process $X(t)$ crosses any given barrier, not necessarily the origin. Several applications can be found in Mathematical Finance (see [1]) and above all in Biology. When one

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 179–186, 2007.
© Springer-Verlag Berlin Heidelberg 2007

considers, for instance, diffusion models for the movement behavior of biological organisms ([4]), one can assume that, in vicinity of a barrier, the individual is able to perceive the presence of two different habitat types and so it will bias its movement towards the preferred habitat type. This gives rise to a movement subjected to a diffusion whose infinitesimal coefficients vary according to which side (with respect to the barrier) is actually occupied by the individual.

We consider the case when $\mu^{\pm}$ and $\sigma^{\pm}$ are linear functions of x, i.e. $\mu^{\pm}(x) = \theta_1^{\pm} x + \theta_2^{\pm},\ \sigma^{\pm}(x) = \eta_1^{\pm} x + \eta_2^{\pm}$. This case includes e.g. BM with drift μ^+ and diffusion coefficient σ^+ when $X(t) > 0$, and μ^- and σ^- when $X(t) < 0$ (i.e. $\theta_1^{\pm} = 0,\ \theta_2^{\pm} = \mu^{\pm},\ \eta_1^{\pm} = 0,\ \eta_2^{\pm} = \sigma^{\pm}$) and the Ornstein- Ulenbeck (O-U) process with parameters $\mu^{\pm} < 0$ and $\sigma^{\pm}$ (i.e. $\theta_1^{\pm} = \mu^{\pm},\ \theta_2^{\pm} = 0,\ \eta_1^{\pm} = 0,\ \eta_2^{\pm} = \sigma^{\pm}$).

For $a, b > 0$, we are interested to study the first-exit time of $X(t)$ from the interval $I = (-b, a)$, when starting at $x \in I$, that is:

$$\tau(x) = \inf\{t > 0 : X(t) = a \text{ or } X(t) = -b | X(0) = x\} \tag{2}$$

and the probability of exit through the right end $\pi^+(x) = P(X(\tau(x)) = a|X(0) = x)$, and through the left end $\pi^-(x) = P(X(\tau(x)) = -b|X(0) = x) = 1 - \pi^+(x)$. For ordinary diffusions (i.e. when $\mu^{\pm}(x) = \mu(x),\ \sigma^{\pm}(x) = \sigma(x)$ are regular enough functions and there is no reflection at zero), explicit formulae are available to calculate $E(\tau(x))$ and $\pi^{\pm}(x),\ x \in (-b, a)$, in the case when $P(\tau(x) < \infty) = 1$ (see e.g. [5], [6]). Lefebvre ([1]) studied the process $X(t) = \mu t + \sigma B_t$ (i.e. asymmetric Brownian motion with drift) where μ and σ depend on the sign of $X(t)$, and jumps of amplitude $\pm\delta$ occur with probability p and $1 - p$ when $X(t) = 0$. For this process he found explicit formulae for $\pi^{\pm}(x)$ and for the moment generating function of $\tau(x)$, in the limit $\delta \to 0^+$.

Although it is interesting to extend the approach of [1] to diffusions more general than BM with drift (such as the O-U process), the calculations involved appear to be very complicated (see [7]). For this reason, in the present article we propose a method to find an approximate solution to the problem; it is obtained by studying the first-passage problem of a suitable process $\hat{X}(t)$ associated to $X(t)$, i.e. a sign-dependent diffusion without reflection at zero, whose infinitesimal coefficients are modified in a neighbor of the origin, in order to take into account the allowed jumps of $X(t)$ of amplitude $\pm\delta$, when it hits zero. In particular, in the case of asymmetric BM with drift, we compare our approximate results for the expectation of $\tau(x)$ and $\pi^{\pm}(x)$ with the exact values found in [1].

2 The First-Passage Time for Asymmetric Diffusions

For $a, b > 0$, let $X(t)$ be the asymmetric diffusion process defined over $I = (-b, a)$, which is the solution of the stochastic differential equation (1). For ordinary diffusions, that is when $\mu^{\pm}(x) = \mu(x),\ \sigma^{\pm}(x) = \sigma(x)$ are regular enough functions and there is no reflection at zero (i.e. $\delta = 0$), explicit formulae are available to calculate $E(\tau(x)),\ \pi^{\pm}(x),\ x \in (-b, a)$ and also the conditional mean exit times at the ends of I, $E(\tau^-(x)) = E(\tau(x)|X(\tau(x)) = -b)$, and $E(\tau^+(x)) = E(\tau(x)|X(\tau(x)) = a)$ (see e.g. [5], [6]). Indeed, assuming the usual

conditions on the regularity of $\mu(x)$ and $\sigma(x)$ (see e.g. [8]), let us denote by L the infinitesimal generator of the process $X(t)$, acting on C^2 functions, i.e. $Lf(x) = \frac{1}{2}\sigma^2(x)f''(x) + \mu(x)f'(x)$. Then, the mean exit time from the interval I of the process started at $x \in I$, say $E(\tau(x))$, is the solution of the problem:

$$Lv(x) = -1,\ x \in I;\ v(-b) = v(a) = 0 \tag{3}$$

while $\pi^{\pm}(x)$ solves the equation:

$$L\pi^{\pm}(x) = 0,\ x \in I;\ \pi^{+}(a) = 1,\ \pi^{+}(-b) = 0\ (\pi^{-}(a) = 0,\ \pi^{-}(-b) = 1)\ . \tag{4}$$

Moreover, $E(\tau^{\pm}(x)) = \frac{T^{\pm}(x)}{\pi^{\pm}(x)}$, where $T^{\pm}(x)$ is the solution of the problem:

$$Lz(x) = -\pi^{\pm}(x)\ ,\ x \in I;\ z(-b) = z(a) = 0 \tag{5}$$

and it results

$$E(\tau(x)) = E(\tau^{-}(x))\pi^{-}(x) + E(\tau^{+}(x))\pi^{+}(x)\ . \tag{6}$$

For the asymmetric BM with drift (see Section 2.1), the explicit formulae for $\pi^{\pm}(x)$ and for the moment generating function of $\tau(x)$ were found in [1], in the limit $\delta \to 0^{+}$. In this paper, our aim is to obtain analogous results for diffusions more general than BM with drift; since the analytical solution requires calculations very complicated -even in the case of the O-U process- (see [7]), instead of searching a formal solution to the problem, we will propose a method to find an approximate solution to it; this will be obtained by studying the first-passage problem of a suitable process $\hat{X}(t)$ associated to $X(t)$, i.e. a sign-dependent diffusion without reflection at zero, whose infinitesimal coefficients are modified in a neighbor of the origin, in order to take into account the allowed jumps of $X(t)$ of amplitude $\pm\delta$, when $X(t) = 0$.

2.1 Asymmetric Brownian Motion with Drift

Let $X(t) = \mu t + \sigma B_t$ be Brownian motion with drift, having infinitesimal parameters which depend on the sign of $X(t)$, that is: $\mu = \mu^{+}$ and $\sigma = \sigma^{+}$ when $X(t) > 0$, and $\mu = \mu^{-}$ and $\sigma = \sigma^{-}$ when $X(t) < 0$. We suppose that, when the process hits the origin, it jumps to the position $\delta > 0$ with probability $p > 0$ and it jumps to the position $-\delta$ with probability $q = 1 - p$. Following [1], $X(t)$ will be called asymmetric Wiener process. The following result holds for the probability $\pi^{+}(x)$ that the process $X(t)$, starting from $x \in (-b, a)$, exits through the right end ([1]). Let be $\beta^{\pm} = 2\mu^{\pm}/(\sigma^{\pm})^2$ and set

$$\pi_{x,a} = \frac{1 - e^{-\beta^{+}x}}{1 - e^{-\beta^{+}a}},\ \text{if}\ \mu^{+} \neq 0;\ \pi^{x,a} = \frac{x}{a},\ \text{if}\ \mu^{+} = 0\ . \tag{7}$$

Then, for $0 \leq x \leq a$, it holds:

$$\mu^{\pm} \neq 0 \Rightarrow \pi^{+}(x) = \pi_{x,a} + (1 - \pi_{x,a})\frac{p\beta^{+}(1 - e^{\beta^{-}b})}{p\beta^{+}(1 - e^{\beta^{-}b}) + q\beta^{-}(e^{-\beta^{+}a} - 1)} \tag{8}$$

$$\mu^+ = \mu^- = 0 \Rightarrow \pi^+(x) = \frac{x}{a} + \left(1 - \frac{x}{a}\right) \frac{pb}{pb + qa} \ . \tag{8'}$$

For $-b < x < 0$, it results:

$$\pi^+(x) = \nu_{x,0} \pi^+(0) \tag{9}$$

where:

$$\nu_{x,0} = \frac{e^{-\beta^- x} - e^{\beta^- b}}{1 - e^{\beta^- b}}, \text{ if } \mu^- \neq 0; \ \nu_{x,0} = \frac{b + x}{b}, \text{ if } \mu^- = 0 \ . \tag{10}$$

Moreover, some rather complicated formulae are available (see [1]) to calculate the mean exit time $E(\tau(x))$ and the moment generating function $E(e^{\lambda \tau(x)})$.

2.2 The Approximate Solution for General Diffusions

As it is well-known, the existence and uniqueness of the strong solution of the stochastic differential equation (1) require $\mu(\cdot)$ to be Lipschitz continuous and $\sigma(\cdot)$ to be Holder continuous (see e.g [8]). Unfortunately, $\mu(x)$ and $\sigma(x)$ are not necessarily continuous at $x = 0$; in fact they pass from $\mu^-(x)$ and $\sigma^-(x)$, for $x < 0$ to $\mu^+(x)$ and $\sigma^+(x)$, for $x > 0$. Thus, it is necessary to consider a regularization, in order to make $\mu(x)$ and $\sigma(x)$ continuous at $x = 0$. Let us consider, for instance, the drift coefficient $\mu(x)$, and suppose that $\mu(x) = \sum_i \theta_i x^i$, where $\theta_i = \theta_i^+$ if $x > 0$, and $\theta_i = \theta_i^-$ if $x < 0$ i.e. $\mu^{\pm}(x)$ are polynomials in the unknown x with coefficients $\theta_i^{\pm}$. We can interpolate any coefficient θ_i in a neighbor of the origin by means of a continuous piecewise-linear function, in the following way. Take a small $h > 0$ and $\alpha \in (\theta_i^-, \theta_i^+)$ and set:

$$\theta_i = \theta_i^- \text{ if } x < x^*; \ \theta_i = \alpha + \frac{x(\theta_i^+ - \alpha)}{h} \text{ if } x^* \leq x \leq h; \ \theta_i = \theta_i^+ \text{ if } x > h \tag{11}$$

where $x^* \doteq -h(\alpha - \theta_i^-)/(\theta_i^+ - \alpha)$. The discontinuity in the parameter appears as h goes to zero. However, if one seeks only a weak solution to (1), it is not necessary to consider any regularization of the coefficients.

The value α is responsible for the asymmetry between the values of the drift at left and at the right of the origin (for $\alpha = (\theta_i^+ + \theta_i^-)/2$ we have perfect symmetry). If e.g. $\mu^{\pm}(x) = \theta_2^{\pm}$, $\sigma^{\pm}(x) = \sigma = const.$, the parameter α can be used to take into account the mean velocity of the process $X(t)$ near zero. If we set $r = \frac{\alpha - \theta_2^-}{\theta_2^+ - \theta_2^-}$, for $r > 1/2$ the process moves to the right more quickly in the average when $X(t) = 0^+$ than when $X(t) = 0^-$, and vice versa holds for $r < 1/2$, while $r = 1/2$ represents the case when the mean velocity of the process is the same in the left and right neighbor of the origin. Of course, the asymmetry between θ_i^+ and θ_i^- is not able itself to realize the mechanism of jumps. To this end, we consider the following approximation argument.

Let us suppose for a moment that $X(t)$ is a homogeneous diffusion without reflection at the origin, and set $\mu(0) = \lim_{x \to 0^+} \mu^+(x)$, $\sigma(0) = \lim_{x \to 0^+} \sigma^+(x)$; then, for $h \to 0$, we have:

$$X(t + h) = X(t) + \mu(X(t))h + \sigma(X(t)) \Delta B_h + o(h) \tag{12}$$

where $\Delta B_h = B_{t+h} - B_t$ is the increment of the BM over the interval $(t, t+h)$. So, recalling the meaning of the drift $\mu(x)$ and the infinitesimal variance $\sigma^2(x)$, it holds:

$$E[X(t+h) - X(t)|X(t) = x] = \mu(x)h + o(h) \tag{13}$$

and

$$E[(X(t+h) - X(t))^2|X(t) = x] = \sigma^2(x)h + o(h)\ . \tag{14}$$

Now, we introduce the reflection at 0; taking into account the allowed jump $\Delta \in \{-\delta, \delta\}$ of the process $X(t)$ at 0, (with $P(\Delta = \delta) = p$ and $P(\Delta = -\delta) = 1 - p$), in place of (12) we obtain, as $h \to 0$:

$$X(t+h) = X(t) + \mu(X(t))h + \sigma(X(t))\Delta B_h + \Delta \cdot \mathbf{1}_{\{0\}}(X(t)) + o(h) \tag{15}$$

where $\mathbf{1}_{\{0\}}(x) = 1$ if $x = 0$, and 0 otherwise. Moreover, since the mean of the random variable Δ is $(2p-1)\delta$ and its variance is $4p(1-p)\delta^2$, we obtain, on the analogy of (13), (14):

$$E[X(t+h) - X(t)|X(t) = x] = \mu(x)h + (2p-1)\delta \cdot \mathbf{1}_{\{0\}}(x) + o(h) \tag{16}$$

and

$$E[(X(t+h) - X(t))^2|X(t) = x] = \sigma^2(x)h + 4p(1-p)\delta^2 \cdot \mathbf{1}_{\{0\}}(x) + o(h)\ . \tag{17}$$

If we choose $\delta = h$, in the approximation $h \approx 0$, we get that the asymmetric diffusion $X(t)$ with reflection at the origin can be approximated by a diffusion $\hat{X}(t)$ without reflection at zero, having infinitesimal parameters $\hat{\mu}(x) = \mu(x) + (2p-1) \cdot \mathbf{1}_{\{0\}}(x)$, and $\hat{\sigma}^2(x) = \sigma^2(x)$ (note that the term $4p(1-p)\delta^2 \cdot \mathbf{1}_{\{0\}}(x)$ is $o(h^2)$).

Alternatively, we can consider the jump-diffusion process (with jumps allowed only when $X(t) = 0$), having infinitesimal generator $L_1 f \doteq Lf + L_j f$ where the additional "jump" part is given by $L_j f = pf(\delta) + (1-p)f(-\delta) - f(0)$ i.e. $p[f(\delta) - f(0)] + (1-p)[f(-\delta) - f(0)]$. At the first order, as $\delta \to 0$, the last quantity equals $pf'(0)\delta + (1-p)f'(0)(-\delta) = \delta(2p-1)f'(0)$ which agrees with the expression already found for $\hat{\mu}(x)$.

In conclusion, the above arguments show that approximations to $\pi^{\pm}(x)$, $E(\tau(x))$ and $E(\tau^{\pm}(x))$ can be obtained by calculating the analogous quantities $\hat{\pi}^{\pm}(x)$, $\hat{E}(\tau(x))$ and $\hat{E}(\tau^{\pm}(x))$ concerning the process $\hat{X}(t)$. Actually, for a given asymmetric diffusion process $X(t)$ we have calculated them, by solving equations (3), (4) and (5) for the diffusion $\hat{X}(t)$. In practice, the drift of $\hat{X}(t)$ has been taken equal to that of $X(t)$, except in a small neighbor of the origin, where the first drift was augmented of the quantity $(2p-1)\delta$. The numerical computations were carried on by running a FORTRAN program written by us, with $\delta = 0.01$. For asymmetric BM with drift, we have compared our approximations $\hat{\pi}^{+}(x)$ and $\hat{E}(\tau(x))$ with the exact values $\pi^{+}(x)$ and $E(\tau(x))$ found in [1], obtaining a satisfactory agreement.

A comparison among the shapes of the estimates of $\pi^{+}(x)$, obtained by using several approximation methods, is reported in the Figure 1, while Figure 2 shows the shape of several estimates of $E(\tau(x))$.

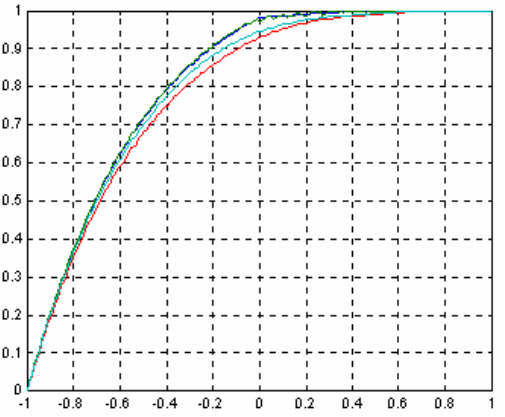


Fig. 1. Comparison among several estimates of $\pi^+(x)$, as a function of $x \in (-b, a)$, for the asymmetric BM with drift, with $a = b = 1,\ \mu^+ = 2,\ \mu^- = 1,\ \sigma = \sigma^+ = \sigma^- = 1$, and $p = 0.8$. From the top to the bottom: exact value $\pi^+(x)$ (*first graph*); $\hat{\pi}^+(x)$ (*second graph*); alternative estimate $\tilde{\pi}^+(x)$, obtained by (4) by interpolating μ near $x = 0$ as in (11) with $h = 0.1$, and $r = 0.8$ (*third graph*), $r = 0.5$ (*fourth graph*).

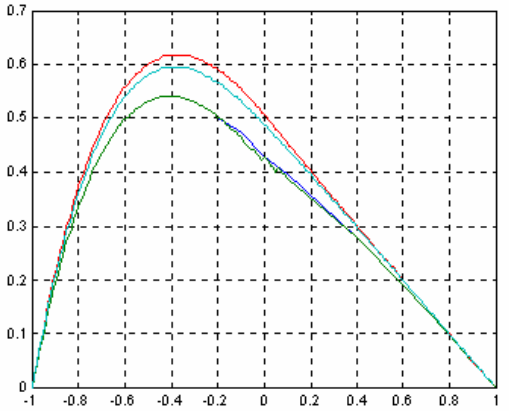


Fig. 2. Comparison among several estimates of the mean exit time from $(-b, a)$ of the asymmetric BM with drift, as a function of $x \in (-b, a)$, for $a = b = 1,\ \mu^+ = 2,\ \mu^- = 1,\ \sigma = \sigma^+ = \sigma^- = 1$ and $p = 0.8$. From the top to the bottom: estimate of $E(\tau(x))$, obtained by (3) by interpolating μ near $x = 0$ as in (11) with $h = 0.1$, and $r = 0.5$ (*first graph*), $r = 0.8$ (*second graph*); the exact value $E(\tau(x))$ (*third graph*) and its approximation $\hat{E}(\tau(x))$ (*fourth graph*). Although $\hat{E}(\tau(x))$ presents some oscillations, the agreement between the third and fourth curve is excellent.

We have also considered the O-U process, i.e. the solution of: $dX(t) = -\mu X(t)dt + \sigma X(t)dB_t,\ X(0) = x \in I\ (\mu, \sigma > 0)$ and the corresponding asymmetric process with reflection at the origin; for both processes we have

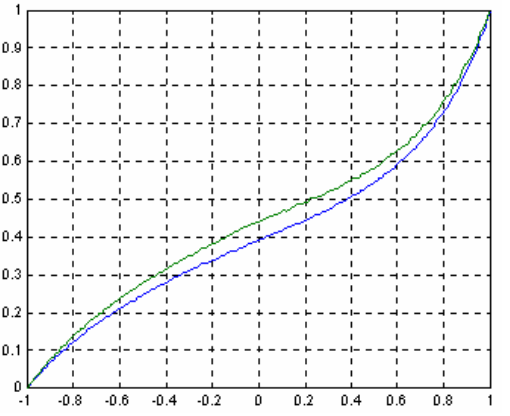


Fig. 3. Comparison among two different estimates of $\pi^+(x)$, as a function of $x \in (-b, a)$, for the asymmetric O-U process which is the solution of $dX(t) = -\mu X(t)dt + \sigma dB_t,\ X(0) = x$, with $a = b = 1,\ \mu^+ = 2,\ \mu^- = 1,\ \sigma = \sigma^+ = \sigma^- = 1$. From the top to the bottom: $\hat{\pi}^+(x)$ for $p = 0.8$ (*first graph*); estimate obtained by (4) by interpolating μ near $x = 0$ as in (11) with $h = 0.1$, and $r = 0.8$ (*second graph*).

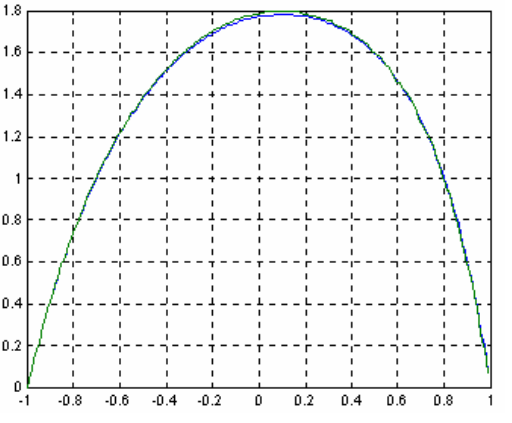


Fig. 4. Comparison among two different estimates of $E(\tau(x))$, as a function of $x \in (-b, a)$, for the asymmetric O-U process which is the solution of $dX(t) = -\mu X(t)dt + \sigma dB_t,\ X(0) = x$, with $a = b = 1,\ \mu^+ = 2,\ \mu^- = 1,\ \sigma = \sigma^+ = \sigma^- = 1$. From the top to the bottom: $\hat{E}(\tau(x))$ for $p = 0.8$ (*first graph*); estimate obtained by (3) by interpolating μ near $x = 0$ as in (11) with $h = 0.1$, and $r = 0.8$ (*second graph*). The curves appear to be appreciably superimposed.

calculated the exit probability through the right end of the interval $I = (-b, a)$ and the mean exit time from I, when starting at $x \in I$ (of course, for the asymmetric process they are only approximate estimates). Figures 3 and 4 show,

respectively, the shape of the estimate of $\pi^+(x)$ and $E(\tau(x))$, obtained by using several approximation methods, for the asymmetric O-U process.

References

1. Lefebvre, M.: First Passage problems for Asymmetric Wiener Processes. J. Appl. Prob. 43, 175–184 (2006)
2. Itô, K., McKean, H.P.: Diffusion Processes and Their Sample Paths. Springer, Berlin (1974)
3. Harrison, J.M., Shepp, L.A.: On Skew Brownian Motion. Ann. Prob. 9(2), 309–313 (1981)
4. Ovaskainen, O., Cornell, S.J.: Biased Movement at a Boundary and Conditional Occupancy Times for Diffusion Processes. J. Appl. Prob. 40, 557–580 (2003)
5. Abundo, M.: On Some Properties of One-Dimensional Diffusion Processes on an Interval. Prob. Math. Statis. 17(2), 235–268 (1997)
6. Karlin, S., Taylor, H.M.: A Second Course in Stochastic Processes. Academic Press, New York (1975)
7. Abundo, M.: First-Passage Problems for Asymmetric Diffusions, and Skew-diffusion Processes (Preprint 2007)
8. Ikeda, N., Watanabe, S.: Stochastic Differential Equations and Diffusion Processes. North-Holland Publishing Company, Amsterdam (1981)

Multivariate Imputation of Genotype Data Using Short and Long Range Disequilibrium

María M. Abad-Grau[1] and Paola Sebastiani[2]

[1] Software Engineering Department, University of Granada, Granada 18071, Spain
mabad@ugr.es
[2] Department of Biostatistics, Boston University, Boston MA 02118, USA
sebas@bu.edu

Abstract. Missing values in genetic data are a common issue. In this paper we explore several machine learning techniques for creating models that can be used to impute the missing genotypes using multiple genetic markers. We map the machine learning techniques to different patterns of transmission and, in particular, we contrast the effect of short and long range disequilibrium between markers. The assumption of short range disequilibrium implies that only physically close genetic variants are informative for reconstructing missing genotypes, while this assumption is relaxed in long range disequilibrium and physically distant genetic variants become informative for imputation. We evaluate the accuracy of a flexible feature selection model that fits both patterns of transmission using six real datasets of single nucleotide polymorphisms (SNP). The results show an increased accuracy compared to standard imputation models. **Supplementary material:** http://bios.ugr.es/missingGenotypes.

Keywords: Bayesian networks, decision trees, imputation, missing data, SNPs, linkage disequilibrium.

1 Introduction

The complete human genome sequence produced by the Human Genome Project has shown that humans share approximately the same DNA with the exception of about 0.1% nucleotide bases [1]. These variations are called *single nucleotide polymorphisms* (SNPs) and occur when a single nucleotide (A, T, C, or G) in the DNA sequence is altered. Some of these SNPs may be markers of DNA sequences that are associated with disease and in the last few years there has been an increased interest in discovering these disease associated polymorphisms. The typical analysis consists of typing the two variants (genotype) in the chromosome pairs of several subjects, and search for those SNPs that are associated with the phenotype –the physical manifestation of the genotype.

Difficulties in typing the genetic markers often lead to missing genotypes and reduced power. Missing genotypes can be imputed and to this end some properties of the human genome can be used to infer, if not reconstruct, the missing data. A key to this process is *linkage disequilibrium* (LD) that is defined as the non-random association between variants of different SNPs in the same

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 187–194, 2007.
© Springer-Verlag Berlin Heidelberg 2007

chromosome. Typically, LD is assumed to hold in sequences of at most 500,000 consecutive nucleotide bases. However, there is evidence that there may be SNPs in LD even at larger physical distance or even different chromosomes. We will refer to this specific situation as *Long Range Disequilibrium* (LRD) to emphasize the lack of physical proximity. In either case, known genotypes of SNPs in disequilibrium with the incomplete one can be used to infer the missing data. This approach is becoming more feasible because with the introduction of technologies for genomewide genotyping, investigators have access to data sets with hundreds of thousands of SNPs.

Motivated by this problem, in this paper we explore several machine learning techniques for creating models that can be used to impute the missing genotypes, using either the LD or LRD assumptions. In the next section we give a brief review of missing data mechanisms and common approaches used to impute the missing data. We describe how learning algorithms can be used as an alternative to simple and multiple imputation methods in Section 2, and we consider the issue of feature selection for the special case of SNPs data sets in Section 3. We describe the evaluation in Section 4. Conclusions and future research are in Section 5.

2 Classifiers as Imputation Models for Discrete Variables

In this paper we will focus on deterministic methods, because they are reproducible and computationally more efficient.

The simplest method is mean substitution that consists of replacing the missing values of a variable by the average of the observed data for that variable. When the variable is categorical, the sample mode is used to fill in the missing data rather than the mean. Imputation based on multivariate models uses information from the other variables in the data set to impute the missing values. Parametric methods use statistical models, such as regression, to fill in each missing value with the value predicted by the model given the observations of the other variables. This method requires building regression models to be used for imputation of each variable with missing data. Non-parametric methods offer an alternative solution to regression-based imputation. An example is hot-deck imputation, in which missing values on a subject are filled in with those of the most similar subject in the data set. This can be considered an application of the 1-nearest neighbor [2], a non-parametric learning algorithm. The main issue of this method is that it can introduce bias when a variable to be completed does not follow any similar pattern in the sample.

We extend here the idea of the hot-deck method and investigate other machine learning methods to build imputation models for discrete variables, as SNPs. Our objective is to leverage on multivariate learning algorithms, particularly those that are robust to missing data and non-relevant variables.

In principle, any algorithm able to learn a classifier from a data set can be used for imputation of the missing values of a discrete variable. In order to do

that, each variable x_i with missing values will be the class variable in a classifier and the remaining variables will be used as input attributes. Therefore, if there are p variables in the sample, p classifiers $c_1, c_2, \ldots, c_i, \ldots, c_p$ will be created, one for each variable. The classifier c_i will be learnt from a subset of the sample in which instances with missing value for the variable x_i are removed. Once the classifier c_i is built, a class value can be obtained given any configurations of the other variables using the decision rule $x_i = c_i(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_p)$. Therefore, the classifier c_i will be used to fill in each missing value of x_i with the value predicted by the classifier given the configuration of the other variables.

K-nearest neighbor classifier is an example that generalizes the hot-deck method from $k = 1$ to $k > 1$ [2]. Some algorithms for building decision trees – tree-shaped constructs of conditions linked to the outcome— have many properties required for imputation of high-dimension samples. In particular, they are robust to non-relevant variables, as they perform an intelligent feature selection. C4.5 is still the reference algorithm to create a decision tree [3]. The algorithm searches for the attributes that are informative to classification and builds a tree that associates each combination of conditions to an outcome. This tree can be used for imputation by filling in each missing value with the class value associated to its combinations of attributes.

Classifiers using Bayesian networks are also very popular [4]. The naive Bayes algorithm is the most common example and assumes independence of the attributes given the class. It can reach very high accuracy despite the extremely simplifying assumption and this may be due to the absence of overfitting, that is common to other more complex Bayesian networks used as classifiers.

3 Feature Selection Using LD and LRD

Not all the variables in a large data set may be useful for imputation, some may be redundant and some may be non informative and their inclusion can decrease the accuracy of classification. Therefore, we need to employ some feature selection algorithms when we build a classifier to be used as model for imputation [5,6].

In the context of genetic models, LD makes some SNPs strongly dependent on each other and this dependency suggests that one can borrow information from SNPs in LD to fill in missing data. At the same time, strong association between SNPs in LD suggests that only a subset of those is necessary for imputation, and that this type of information should be included in feature selection algorithms. Based on either the LD ar LRD assumption we can envision two types of searches: (1) *LD-based search:* the idea here is to search for SNPs to be treated as attributes in the classification model only among those in LD with the class variable, i.e. the SNP with missing data to be filled in and (2) *LRD-based search:* the idea is to relax the assumption that association is possible only between SNPs in LD and to look for attributes to be used in the classification in the whole set.

While the second type of search can be performed with any feature selection algorithm, the first type of search requires first to identify SNPs in LD with the

class variable. There are several algorithms able to divide a sequence of SNPs in *haplotype blocks* – regions with SNPs in strong LD separated by recombination hotspots [7,8]. Once these regions are identified, the LD based search will be limited to SNPs in blocks of LD, and be less computationally heavy. Based on the intuition that closer SNPs are in stronger LD, a simple approach consists of ranking SNPs in a block by their physical distance from the SNP that needs to be imputed, so that the k nearest SNPs are included, for each given k.

However, there is growing evidence that SNP selection methods that do not impose block boundaries, as the "haplotype motif" model [9], may perform better than block-based methods in identifying the necessary associations between SNPs. Moreover, in data sets of individuals with a complex disease determined by several interacting genes, association may exist between far apart SNPs or even between SNPs in different chromosomes, if they are markers close enough to genes involved with the disease. Thus, block detection algorithms may ignore SNPs that are informative for imputation of missing data and may keep redundant SNPs because they are inside a block, while generic algorithms for feature selection used as a preprocessing step for building classifiers may constitute a better approach. While some of these algorithms, such as C4.5, include an embedded method for feature selection, most classifiers require an external feature selection algorithm to improve accuracy in presence of non-relevant input variables. In the wrapper selection, the selection algorithm uses accuracy of the classifier to select input variables, by wrapping the classifier with the selection algorithm. Wrapper methods are computationally more expensive than filter methods –those performing selection as a preprocessing step when learning a classifier– but they achieve better results [5,6].

4 Evaluation

We have used six data sets called TGF, Crohn, Chromosome 1 CEPH, Chromosome 1 Yoruba, Chromosome 22 CEPH and Chromosome 22 Yoruba with l categorical variables, with $l = 49, 104, 100, 100, 100$ and 100 respectively. The first data set is part of a genetic study looking at genetic modulators of disease severity in subjects with sickle cell anemia [10]. The set consists of 1595 instances, 48 SNPs (Table 1) and one binary variable that represents the presence or absence of a vaso-occlusive complication of sickle cell anemia. The Crohn data set is a publicly available set that was originally used in [11]. It consists of the genotype data of 104 SNPs typed in 258 unrelated subjects who are the unrelated parents of 129 trios [12]. The phenotype is the presence/absence of Crohn disease. The SNPs span across 500 kilobases on chromosome 5q31 and the region contains 11 known genes. The other four sets were derived from data publicly available from the website of the International HapMap project [13], and consist of genotype data of 100 SNPs in chromosome 1 and 22, typed in two populations: Yoruba in Ibadan, Nigeria (YRI) and CEPH who are Utah residents with ancestry from northern and western Europe (CEU). The original data consists of the genotypes of 30 trios but we used only the genotypes data of

Table 1. Range of SNPs used in the TGF data set that belongs to the same chromosome (row 1), the number of genes for each range (row 2) and the chromosome (row 3) they belong to

SNPs	1-3	4-15	17-18	19-24	25-27	28-38	39-48
number of genes	1	2	1	1	1	2	4
chromosome	2	4	5	10	13	15	18

Table 2. Range of SNPs used in the data sets from the IHMP

Data set	initial SNP	last SNP	physical distance
chrom 1 CEU	189163057	189254049	90992
chrom 1 Yoruba	202331052	202413027	81975
chrom 22 CEU	16972056	17266765	294709
chrom 22 Yoruba	42514274	42570618	56344

the parents to avoid correlations between the observations. Therefore each of the four sets consists of 60 instances: the number of chromosomes in 30 individuals. We selected the genotypes of 100 adjacent SNPs in randomly selected regions of the two chromosomes.

We removed approximately 20% of known genotype data using a mechanism that produces data that are missing at random. Table 2 shows for each data set the physical position for the first and last SNP chosen and the total physical distance. On average, there is a 10, 25, 20, 20, 20, and 20 percent of missing values in each data set respectively.

We use these methods to impute missing values:

- Mode substitution (MS).– For a given SNP, each missing genotype is replaced by the most-frequent genotype observed in the complete observations.
- Phenotype-based model (PB).– This generalizes the MS method when there is a categorical phenotype so that each missing value is replaced by the most common genotype in the observed data, conditional on the phenotype. This method could only be used with the TGF and Crohn data sets because they are the only ones that include phenotypic information.
- Hot-deck imputation .– This is the k-nearest neighbour classifier that we used for $k = 1$ (1-nn) and $k = 5$ (5-nn).
- Naive Bayes.– The Naive-Bayes classifier was used for imputation of missing data as explained above. Estimation was based on Maximum Likelihood (NB0) and a Bayesian approach with a prior Dirichlet distribution (NB1) [4].
- Decision trees (C4.5).– The C4.5 algorithm was used for building tree-based classifiers to be used for imputation of missing data [3].

Note that when only one variable is used, both NB0 and C4.5 are equivalent to MS. Similarly, PB imputation is equivalent to NB0 and C4.5 when only the

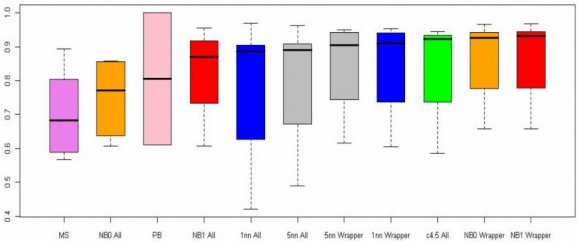


Fig. 1. Box plots displaying the accuracies obtained by the various procedures in all the data sets used. Each box plot shows the results of one algorithm and they are ordered by median accuracy. The word 'all' means that all SNPs were used, while 'wrapper' means a feature selection was first performed. C4.5 already included a feature selection method.

SNP whose missing values are to be imputed and the phenotype attribute are used.

While C4.5 has an embedded mechanism to select attributes, none of the other multivariate algorithms does. Therefore, we used the naive Bayes classifier and the nearest-neighbour classifier using all the variables in the data set, and also applied a non-positional feature selection algorithm. The algorithm is a wrapper forward feature selection that starts with an empty set of attributes, adds to the current set the attribute that, together with the current set, obtains the highest accuracy, and continues until there is no attribute left that would increase the accuracy if included in the classifier. Note that neither the algorithm embedded in C4.5 nor the non-positional feature selection algorithm impose any restriction in the selection of attributes, so that they naturally fit the LRD patterns of association but they can be adapted to the LD patterns of association, if the selection is applied to the set of SNPs in LD with the SNP used as class variable.

Figure 1 shows the results of our evaluation when we used the LRD assumption to select SNPs to be included in the classifiers. Each boxplot displays the average accuracies of the different methods applied to impute missing data for each SNPs of each data set. The accuracy of different methods was estimated by the number of imputed genotypes that matched the original ones and by using 5-fold cross-validation. The results show a clear superiority of multivariate algorithms compared to the simple mode imputation (median accuracy of mode imputation is less than 70%, compared to median accuracies of NB0, NB1, 1-nn, and 5-nn in the range 79%-89%). Furthermore, multivariate algorithms built with some feature selection mechanism have a significant improvement in accuracy compared to multivariate algorithms that use blindly all the variables in the data set. The Naive Bayes classifier with feature selection has the best performance, with a median accuracy greater than 90%. The accuracy of C4.5 (median value 90%) is slightly inferior to that of the Naive Bayes classifier and, given the considerable computational effort of the feature selection algorithm, it is a valuable alternative.

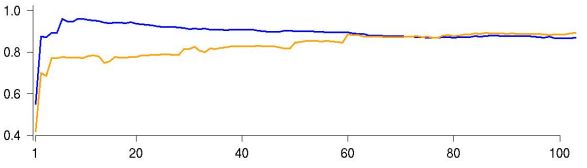


Fig. 2. Mean accuracies (y axis) for imputation of missing data in SNP 29 (blue line, LD pattern) and SNP 49 (orange line, LRD pattern) of the Crohn data set, considering an increasing block size (x axis)

Figure 2 shows an example of feature selection based on the LD assumption using two SNPs of the Crohn data set. We evaluated the accuracy of all methods, using an increasing number of SNPs that were selected based on physical distance. The x-axis displays the block size and the y-axis the average accuracy of all the algorithms used for each block. Therefore using blocks of increasing size in the imputation model can lead to either an increase (SNP 29, orange line) or decrease (SNP 49, blue line) of accuracy and this may be due to the fact that the SNP 29 is in LD with only SNPs within a small distance while the SNP 49 is in LD with SNPs that are further away. For lack of space, other examples can be seen at "http://bios.ugr.es/missingGenotype/index1.php" and suggest that those algorithms fitting both patterns will lead to more reliable results than those using blocks, i.e., assuming LD.

5 Conclusions

In this paper we have shown that algorithms to learn classifiers can be successfully applied to impute missing data. Our results show that the use of learning algorithms with feature selection can fill in the missing values with more than 90% accuracy compared to algorithms that use all the variables and reach median accuracies in the range 79%-89% leads to choosing the informative variables for imputation and increases the accuracy of the imputed data.

The surprising result of our analysis is that restricting attention to LD rather than LRD decreases the accuracy of the imputation models. Because the size of genetic data increases at exponential rate and common genome wide association studies produce hundreds of thousands of variables, the ability to restrict feature selection to a subset of the SNPs would result is a substantial saving of computation. In future work we propose to improve the algorithms that we analyzed here by using current knowledge about physical distance between SNPs.

Acknowledgements. Research supported by NHLBI grants R21 HL080463-01 and 1R01DK069646-01A1 and the Spanish Research Program under projects TIN2004-06204-C03-02 and TIN2005-02516.

References

1. Consortium, T.G.I.S.: Initial sequencing and analysis of the human genome. Nature 409, 860–921 (2001)
2. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithm. Machine Learning 6, 37–66 (1991)
3. Quinlan, J.R.: Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research 4, 77–90 (1996)
4. Sebastiani, P., Abad-Grau, M.M., Ramoni, M.F.: Learning Bayesian Networks. In: Maimon, O., Rokach, L. (eds.) Data mining and knowledge discovery handbook, pp. 193–230. Springer, Heidelberg (2005)
5. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant features and the subset selection problem. In: Proceedings of the Eleventh International Conference on Machine Learning, pp. 121–129. Morgan Kaufmann, San Francisco (1994)
6. Kohavi, R., John, G.H.: The wrapper approach. In: Artificial Intelligence Journal, Springer, Heidelberg (1998)
7. Patil, N., Berno, A., Hinds, D., Barrett, W., Doshi, J., Hacker, C., Kautzer, C., Lee, D., Marjoribanks, C., McDonough, D.: Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. Science 294 (2001)
8. Gabriel, S., Schaffner, S., Nguyen, H., Moore, J., Roy, J., Blumenstiel, B., Higgins, J., DeFelice, M., Lochner, A., Faggart, M., Liu-Cordero, S.N., Rotimi, C., Adeyemo, A., Cooper, R., Ward, R., Lander, E., Daly, M., Altshuler, D.: The structure of haplotype blocks in the human genome. Science 296 (2002)
9. Castellana, N., Dhamdhere, K., Sridhar, S., Schwartz, R.: Relaxing haplotype block models for association testing. In: Proceedings of the Pacific Symposium on Biocomputing, vol. 11, pp. 454–466 (2006)
10. Baldwin, C.T., Nolan, V.G., Wyszynski, D.F., Ma, Q.L., Sebastiani, P., Embury, S.H., Bisbee, A., Farrell, J., Farrer, L.S., Steinberg, M.H.: Association of klotho, bone morphogenic protein 6 and annexin a2 polymorphisms with sickle cell osteonecrosis. Blood 106(1), 372–375 (2005)
11. John, D., Rioux, M.J.D., Silverberg, M.S., Lindblad, K., Steinhart, H., Cohen, Z., Delmonte, T., Kocher, K., Miller, K., Guschwan, S., Kulbokas, E.J., O'Leary, S., Winchester, E., Dewar, K., Green, T., Stone, V., Chow, C., Cohen, A., Langelier, D., Lapointe, G., Gaudet, D., Faith, J., Branco, N., Bull, S.B., McLeod, R.S., Griffiths, A.M., Bitton, A., Greenberg, G.R., Lander, E.S., Siminovitch, K.A., Hudson, T.J.: Genetic variation in the 5q31 cytokine gene cluster confers susceptibility to crohn disease. Nature Genetics 29, 223–228 (2001)
12. Daly, M.J., Rioux, J.D., Schaffner, S.F., Hudson, T.J., Lander, E.S.: High-resolution haplotype structure in the human genome. Nat. Genet. 29, 229–232 (2001)
13. HapMap-Consortium, T.I.: The international hapmap project. Nature 426, 789–796 (2003)

Neonatal EEG Sleep Stages Modelling by Temporal Profiles

Vladimír Krajča[1], Svojmil Petránek[1], Jitka Mohylová[2], Karel Paul[3], Václav Gerla[4] and Lenka Lhotská[4]

[1] Department of Neurology, Faculty Hospital Na Bulovce, Prague, Czech Republic
[2] VŠB-Technical University, Ostrava
[3] Institute for Care of Mother and Child, Prague
[4] Czech Technical University, Prague
{krajcav@fnb.cz}

Abstract. The paper deals with the application of the EEG temporal profiles for the neonatal sleep stages modelling. The temporal profiles created by adaptive segmentation and cluster analysis reflect the time structure of the EEG during different periods of sleep. They can be used for neonatal EEG quantification and for the detection of sleep stage changes.

Keywords: EEG, neonates, sleep stages, adaptive segmentation, cluster analysis, temporal profile.

1 Introduction

The electroencephalogram (EEG) is an important tool in neonatal brain maturation assessing. For visual evaluation it is necessary to track the percentual changes of a different EEG activity – burst and interburst periods, different segments duration, and other signal parameters. Evaluation of such an EEG record is time consuming and requires an experienced investigator. The EEG signal features depend strongly on sleep stage of the infant. From this reason, the sleep stage identification is very desirable. In this paper, we would like to show, that sleep structure of the EEG can be modeled by structural time profiles and that the sleep stage changes are reflected in the EEG segment class membership changing.

1.1 A Phenomenological Model Based on Adaptive Segmentation

Bodenstein and Praetorius [1] were the first who suggested a phenomenological model for the EEG. The signal was split into elementary patterns by adaptive segmentation with the help of a fixed and a moving window. The method did not allow the simultaneous independent segmentation of multichannel signals. From the patterns the spectral features were extracted and used for segments classification by a hierarchical cluster analysis.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 195 – 201, 2007.
© Springer-Verlag Berlin Heidelberg 2007

1.2 Temporal Profile

Temporal profile shows the class membership of the segments in different segment classes in the course of time. Barlow [2] was the first who explored the possibility to use the temporal profiles for the characterization of tracé alternant and REM sleep patterns in neonatal EEG. The plotting of profiles was performed on a single channel, and spectral and statistical parameters of the segment duration were computed, without any further analysis of the time structure of the temporal profiles. The analysis was performed within particular state (REM or trace alternant) without any further attempt to use it for differentiation of the states.

2 Materials

A total of 21 healthy sleeping newborns (10 full terms and 11 preterms) were recorded polygraphically (EEG – 8 leads, respiration, ECG, EOG, and EMG) in standard conditions using a paperless EEG recording system for 90-120 minutes. An experienced physician evaluated the records. The periods of quiet sleep (QS) and active sleep (AS) were marked by flags during visual inspection of the EEG recordings [4]. Artifacts were also marked during recordings by an EEG technician.

The examinations were carried out in an EEG laboratory in standardized conditions after morning feeding and lasted 90-120 minutes. The EEG activity was recorded polygraphically from 8 derivations, positioned under the system 10-20 (Fp1, Fp2, C3, C4, O1, O2, T3, T4); reference derivation: linked ear electrodes; filter setting: 0,4 Hz and 50 Hz. The respiration (PNG), ECG, EOG, and EMG of chin muscles were also recorded. Electrode impedances were not higher than 5 kOhms. The recording was performed using the Brain-Quick (Micromed) digital system with a sampling frequency of 128 Hz. An observer continuously recorded any change in the infant's behavior on the polygram.

3 Methods

For sleep stage assessment, it is necessary to find the beginning and the end of relevant graphoelements, to classify (identify) them and to compute the percentual occurrence of the waves. It can be done by adaptive segmentation, feature extraction and classification of the segments by a cluster analysis method. The signal statistics can be computed from the classes. The class membership in the course of time can be plotted as the structural time profile.

In this paper, we will show, that multichannel temporal profiles can be used for modelling the time structure of different stages of the neonatal sleep EEG and they can be processed to provide the detection curve, which can be applied for change of sleep state detection.

3.1 Multichannel Adaptive Segmentation and Feature Extraction

To take into account the non stationary behaviour of the signal, the features for classification were extracted from piece-wise stationary segments detected by an adaptive segmentation algorithm. This method divides the signal into quasi-stationary (piece-wise stationary) segments depending on the change of the stationarity of the signal.

For adaptive segmentation, the method utilizing two connected windows was used [3]. The two windows slide along the signal. The change of stationarity is indicated by local maxima of their difference measure (combined amplitude and frequency difference). The small fluctuations of the difference measure are limited by a threshold. The principle of the method is shown in Fig. 1.

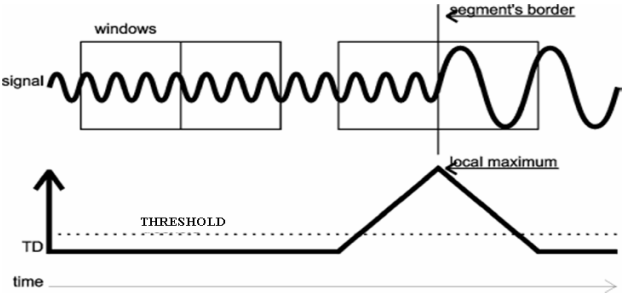


Fig. 1. The principle of adaptive segmentation using the two connected windows and a simple difference measures (mean frequency and mean amplitude of the windows)

Feature Extraction. Numerical data were obtained for each EEG lead from each sample. The features were the following: the standard deviation of the sample values in the segment, the difference between maximal positive and minimal negative values of the samples in the segment, maximum of absolute values of the first derivative of the samples in the segment, maximum of absolute values of the second derivative of the samples in the segment, average frequency of EEG activity in the segment ,and square root from power spectral density (PSD) in the frequency bands: delta (0,5 –3 Hz), theta (3-8 Hz), alpha (8-12 Hz) and beta. (12-30 Hz) [4],[5]. The PSD were calculated using Fast Fourier Transform.

3.2 Cluster Analysis and Temporal Profiles

The classification was performed using cluster analysis (fuzzy c-means algorithm [6]). The multichannel time profiles – the functions of class membership depending on time - were plotted for each analyzed EEG channel to reflect the temporal structure of the signal [7],[8]. An example is shown in Fig.2. The temporal profile reflects the dynamic EEG structure during sleep and can be used for the sleep state change detection.

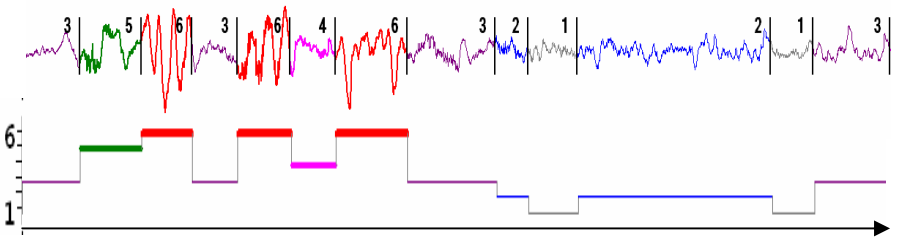


Fig. 2. Example of temporal profile of one channel of the EEG; temporal profile (below) corresponds to the segments identified by adaptive segmentation and cluster analysis (above). Numbers indicate the class membership, vertical lines mark the segment boundary and the different classes are distinguished by different colors.

3.3 Temporal Profiles Processing

The automatic sleep detection is very desirable for functional brain maturation study in neonates, as discussed in [9], where sleep stages are determined visually. Further on, for objective quantitative analysis of the maturation of the child brain, we need precise signal statistics.

Adaptive segmentation can contribute to this purpose by finding the exact instants of burst/interburst events and changes of signal stationarity by dividing the signal into quasi-stationary parts [2].

Both aforementioned tasks, i.e. sleep stages detection and quantitative parameters extraction, can be made by structural analysis of the EEG temporal profiles. The temporal profiles can serve for the finding the instants of sleep stages changes and the detailed statistical analysis can be done by computing the percentual occurrence of different EEG graphoelements detected by adaptive segmentation and which can be labeled by cluster analysis. The purpose of this study was to show, how the structural time profiles analysis can provide a unified approach to all the above mentioned tasks.

The processing of time profiles consisted of the following steps (see Figure 3.):

1) Multichannel time profiles were created by adaptive segmentation and cluster analysis.
2) The class membership in time profiles was averaged to obtain a single detection curve from all channels. Running mean of averaged class membership was computed by short moving average (MA) filter (10 points).
3) Class membership was subtracted from the running mean and squared.
4) Resulting curve was smoothed with the help of another moving average filter (50 points).
5) The threshold for the change of sleep stage detection was computed as an average of the final curve.

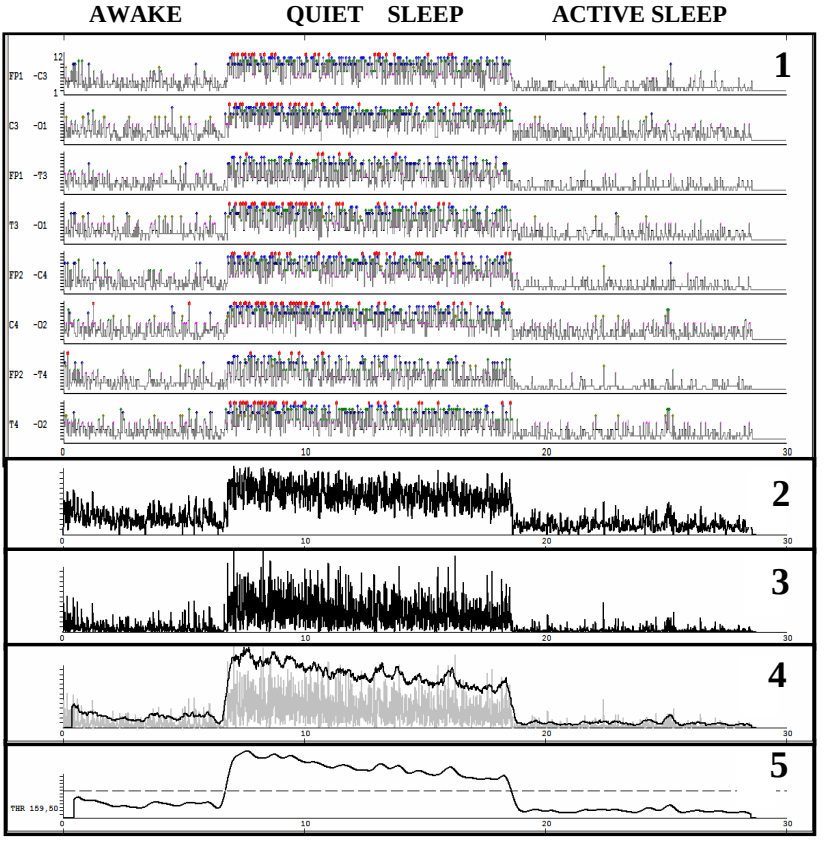


Fig. 3. Processing of the time profile. 1- Multichannel time profiles (8 EEG channels). 2-averaged channels. 3-the squared difference from running mean. 4-smoothed difference. 5-final smoothing and comparison to the threshold (mean of the curve).

4 Results

An example of structural description of neonatal EEGs is shown in Fig. 4. The awake period and quiet and active sleep stages may be clearly distinguished. Eight EEG channels are plotted. The signal graphoelements were classified into 6 classes reflecting the occurrence of the basic EEG graphoelements in the course of time. The different sleep stages, as evaluated by visual inspection of the record by physician, are indicated in the bottom of the diagram. The relevant classes of the EEG graphoelements may be identified using different colors to provide the clear graphic presentation of time structure of the neonatal sleep EEG. This structure was apparent in all analyzed recordings.

Artefacts. If the processed temporal profiles should be used not only for the modelling of the time structure of the EEG recordings, but also for the sleep stage

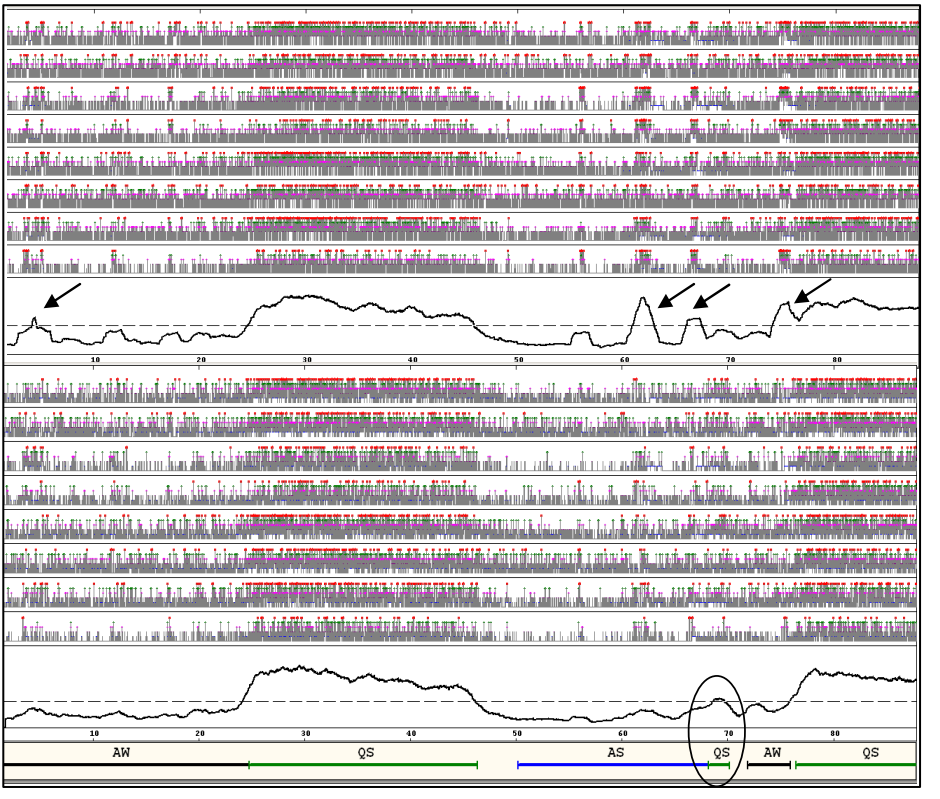


Fig. 4. Detection of different sleep stages by processing of the time profile. AW-awake, QS-quiet sleep, AS-active sleep. The upper part – negative influence of artefacts on the detection curve (arrows indicate the artefact activity). After simple method of moving artefact elimination, the quality of the detection curve is improved and even very short and transient quiet sleep stage is detected (ellipse).

changing detection, the influence of artefacts, that might occur in the signal must be taken into consideration. In the upper part of the Fig. 4. there is shown the EEG signal processed without artefact elimination. There can be seen, that artefact activity provides the false indication of detection curve crossing (it is marked by arrows).

In the bottom part of the figure there is the same EEG signal processed after simple artifact elimination : all high voltage activity (with the absolute value above 150 microvolts) is eliminated from the signal – the signal samples are set to zero. The quality of detection curve is improved.

5 Conclusions

We developed a new method for automatic sleep stages detection in neonatal EEG based on time profiles processing and we applied it to fullterm neonatal EEGs.

The initial tests proved that the method is sufficiently sensitive to detect the sleep stages in the EEG of infants. The method is based on adaptive segmentation, feature extraction and subsequent classification by cluster analysis. The resulting structural time profiles are processed by a novel algorithm to reveal the sleep stages in preterm infant EEG. The signal statistics can be also computed during multichannel signal processing and quantitative features and parameters can be extracted from the classified EEG segments.

The study showed that the modelling of the sleep structure of the multichannel EEG of the newborns by time profile matches the visual evaluation of the physician and provides a good basis for fully automatized method of sleep stages identification.

Acknowledgments

This work was supported by grants IGA MZ ČR 1A8600 and AV ČR 1ET101210512.

References

1. Bodenstein, G., Praetorius, H.M.: Feature extraction from the Electroencephalogram by adaptive segmentation. Proc. of the IEEE 65(5), 642–652 (1977)
2. Barlow, J.S.: Computer characterization of trace´ alternant and REM patterns in the neonatal EEG by adaptive segmentation-an exploratory study. Electroenceph Clin. Neurophysiol. 60, 163–173 (1985)
3. Krajca, V., Petranek, S., Pataková, I., Varri, A.: Automatic identification of significant graphoelements in multi-channel EEG recordings by adaptive segmentation and fuzzy clustering. Int. J. Biomed. Compute. 28, 71–89 (1991)
4. Paul, K., Krajca, V., Roth, Z., Melichar, J., Petranek, S.: Comparison of quantitative EEG characteristics of quiet and active sleep in newborns. Sleep Medicine 4, 543–552 (2003)
5. Paul, K., Krajča, V., Roth, Z., Melichar, J., Petránek, S.: Quantitative topographic differentiation of the neonatal EEG. Clinical Neurophysiology 117, 2050–2058 (2006)
6. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Functions Algorithms. Plenum Press, New York (1981)
7. Krajča, V., Mohylová, J., Paul, K., Petránek, S.: Automatic detection of sleep stages in preterm neonates by exploring the time structure of EEG, The 3rd European Medical and Biological Conference EMBEC 2005 (November 20-25, 2005), Prague (2005)
8. Krajca, V., Petranek, S., Paul, K., Matousek, M., Mohylova, J., Lhotska, L.: Automatic Detection of Sleep Stages in Neonatal EEG Using the Structural Time Profiles. In: EMBC05, Proceedings of the 27th Annual International Conference of the IEEE-EMBS (September 1-4, 2005), Shanghai, China (2005)
9. Scher, M.S., Jones, B.L., Steppe, D.A., Cork, D.L., Seltman, H.J., Banks, D.L.: Functional brain maturation in neonates as measured by EEG-sleep analyses. Clin. Neurophysiol. 114, 875–882 (2003)

On the Representation of Imperative Programs in a Logical Framework*

José Luis Freire Nistal, Enrique Freire Brañas, Antonio Blanco Ferro, and David Cabrero Souto

University of A Coruña, Spain
{freire, efreire, blanco, cabrero}@udc.es
University of A Coruña, LFCIA, Dept. of Computer Science, Faculty of Informatics, 15071 A Coruña, Spain

Abstract. Research on formal verification of imperative programs using some form of representing them in a type theory has been done for years. Generally, the different approaches include a verification conditions generator, which from an annotated program including variants and invariants for while–loops and using a Hoare logic–like specification, produces some propositions to be proved in a logical framework, expressing the program correctness and termination.

In this paper we present a direct use of Coq [3] to model imperative programs. This method, and the fact that it is not possible to have not–ending programs in Coq, should allow a more deep understanding of imperative programs semantics [15], and people without big knowledge on type theory could use this theorem prover to verify imperative programs properties. This approach is based on using a fixed–point equality theorem [2] that represents the appropriate reduction rule to be used in our model.

In our approach no Hoare logic rules are used for verification of program specifications. This verification is achieved, in a pure constructive setting, directly with the type theory model.

1 Introduction

We present a direct use of Coq on modeling imperative programs. This method, and the fact that it is not possible to have not–ending programs in Coq, should allow a more deep understanding of imperative programs semantics, and people without big knowledge on type theory could use this theorem prover to verify imperative programs properties.

The fundamental ideas to model variable assignment and memory on a type theory are well known [14], [10]. In [4] operational and denotational semantics of a simple imperative language are formalized in Coq and also a formal proof of their equivalence is given. We follow these basic ideas to implement variable assignment and memory states as it is shown in section 3. One of the most complete

* Supported by: MEC, Spain, Project TIN2005-08986. FARMHANDS.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 202–209, 2007.
© Springer-Verlag Berlin Heidelberg 2007

and promising approaches is the system Why [8], a verification conditions generator: from an annotated C or Java program including variants and invariants for while–loops, written in a intermediate language and using a Hoare logic–like specification, it generates proof obligations for Coq (and some other theorem provers) expressing the correctness and termination of the given program.

In this paper a less powerfull but (we hope) still useful method is proposed. Due to space limitations and in order to show as clear as possible the proposed metodology, we will focus on programs containing only variable assignation and `while` loops. For that, we use the Coq facility for building non structural recursive functions using well–founded recursion. Classically this include making structural recursion on the proof structure that there is no infinite descending chain starting from the argument. This recursion can be rather difficult for those with no deep knowledge of type theory. Instead that, we will use a fixed–point equality theorem [2] that represents the appropriate reduction rule to be used when proving program properties. A simple example is included to illustrate how our methodology can be applied in practise.

For the actual Coq 8.1 code of this specific development see the following address: `www.dc.fi.udc.es/staff/freire/publications/publications.shtml`

2 Background

The Coq logical framework is an implementation of the Calculus of Inductive Constructions by G. Huet, T. Coquand and C. Paulin–Mohring developed at INRIA. It is a goal-directed and tactic–driven theorem prover following the judgments–as–types representation paradigm [11], [13], [7].

It features a set of predefined tactics, including an `auto` tactic which tries to apply previous proofs. The default logic is intuitionist but classical logic is also available by importing the `Classical` module.

The notation `a:A` (`a` is of type `A`) is interpreted as "`a` is a proof of `A`" when `A` is of type `Prop`, or "`a` is an element of the specification `A`" when `A` is of type `Set`. Here, `Prop` and `Set` are the basic types of the system. These two types and a hierarchy of universes `Type`(i) for any natural i are the elements of the set of sorts with the rules: `Prop:Type`(0) and `Type`(i)`:Type`$(i+1)$.

Allowed constructions are: `x| M N |fun (x:T)=>f|forall x:T,U`, where `x` denotes variables as well as constants; `M N` is the application; the third expression represents the program (λ–expression) with parameter `x` and term `f` as body ($\lambda(x : T) \cdot f$, the abstraction of variable `x` of type `T` in `f`). Finally, the fourth is the type of programs that admit an entry of type `T` and return a result of type `U`. It is referred to as product type, usually represented in type theory by $\Pi(x : T) \cdot U$ or also $\forall(x : T) \cdot U$. If `x` is not free in `U`, then this is simply written $T \rightarrow U$.

Typing rules provide also proof tactics when reading bottom up. For example

$$\frac{E[\Gamma] \vdash \forall(x : T) \cdot U : s \quad E[\Gamma :: (x : T)] \vdash f : U}{E[\Gamma] \vdash \lambda(x : T) \cdot f : \forall(x : T) \cdot U}$$

expresses that the term (program) $\lambda(x:T)\cdot f$ has the product type $\forall(x:T)\cdot U$ provided that this type has some sort type and term f has type U in the environment E and context Γ if $x:T$. Therefore, given the type `forall x:T,U` in Coq, if one looks for some term with this type, the `intro` tactic can be used, which leads to the subgoal `U` because if we can construct `f` of type `U` then Coq itself builds the term `fun (x:T)=>f`.

Inductive types can also be defined in the system Coq and to each of them corresponds an structural induction principle, and possibly a recursion scheme to build programs, automatically generated by the system. For instance, the type of natural numbers can be defined in a Coq session[1] and also some properties of its recursive scheme checked as follows:

```
Coq < Inductive nat:Set := O:nat | S:nat -> nat.
nat is defined
nat_ind is defined
nat_rec is defined
Coq < Parameters P:nat->Set;o:(P O);h: forall n,(P n)->(P (S n));n:nat.
Coq < Eval compute in (nat_rec P o h O).
     = o
     : (P O)
Coq < Eval compute in (nat_rec P o h (S n)).
     = (h n (nat_rec P o h n))
     : (P (S n))
```

Other types, as binary integers `Z` and logic properties can also be defined as inductive types. A good introductory book on Coq is [3].

In order to ensure decidability of type-checking (and hence of proof-checking), proof assistants based on type theory require that functions are provably terminating, total, and deterministic. Totality and determinacy are enforced by requiring definitions by pattern matching to be exhaustive and unambiguous, and termination is enforced through a guard criterion that checks that recursive calls are performed on structurally smaller arguments. Nevertheless Coq allows also the use of non–structural orders whenever it is proved that they are well–founded [2], [5]. This facility has been largely improved in the actual version 8.1.

3 The Very Basic Steps

To define variable assignment we start with a equality–decidable type for variables. Then memory–states, as functions that can be updated, are defined:

```
Coq < Variable Var:Set.
Var is assumed
Coq < Variable d:forall x y:Var,{x=y}+{~(x=y)}.
d is assumed
Coq < Let States:= Var->Z.
States is defined
```

[1] The typewriter font indicates input code and the slanted text corresponds to the output in an interactive session. All input phrases end with a dot.

```
Coq < Definition update (x:Var) (n:Z) (s:Var->Z) := (fun (y:Var)=>
Coq <                    if (d x y) then n else (s y)).
update is defined
```

4 The `while-do` Problem

In order to model the `while p do f` loop construct we will only consider in this paper the most simple scheme possible. We need a function for the loop–body $f : X \to X$, a relation $R : X \to X \to Prop$ and a predicate for the test $P : X \to Prop$. This predicate must be decidable and, for ensuring termination, the property $\forall x \in X \cdot (P\,x) \Rightarrow (R\,(f\,x)\,x)$ must be true. In Coq we will have the following context:

```
Coq < Variable X:Set.
Coq < Variable R:X->X->Prop.
Coq < Variable P:X->Prop.
Coq < Variable r:forall x, {P x}+{~ (P x)}.
Coq < Variable f:X->X.
Coq < Variable term:forall x:X,(P x)->(R (f x) x).
```

with iteration as a structural recursive function:

```
Coq < Fixpoint itern (n:nat):X->X:=
Coq < match n with
Coq < 0%nat => fun x:X => x
Coq < |S m => fun x:X => f (itern m x)
Coq < end.
```

5 Well–Founded Relations

Let $\prec$ be a binary relation on a set X and $Fin(X, \prec)$ the set of elements $a \in X$ such that there is no infinite descending sequence $\{a_n\}_{n \in \mathbf{N}}$ verifying

$$\ldots a_{n+1} \prec a_n \prec \ldots a_2 \prec a_1 \prec a.$$

then $\prec$ is called well–founded if $X = Fin(X, \prec)$.

Furthermore, the concept of accessibility can be defined [1] as an inductive predicate: an element $x \in X$ is accessible if every $y \in X$ such that $y \prec x$ is accessible:

$$\forall x : X \cdot \big(\forall y : X \cdot x \prec y \Rightarrow (Acc \prec y)\big) \Rightarrow (Acc \prec x)$$

obtaining that $\prec$ is well–founded if and only if $X = Acc(X, \prec)$.

These concepts are defined in the basic Coq library and they can be shown, if we represent $\prec$ as `R:X -> X -> Prop`, as follows:

```
Coq < Print Acc.
Inductive Acc (X : Set) (R : X -> X -> Prop) : X -> Prop :=
    Acc_intro : forall x : X,
```

```
                (forall y : X, R y x -> Acc R y) -> Acc R x
Coq < Print well_founded.
well_founded =
fun (X : Set) (R : X -> X -> Prop) => forall x : X, Acc R  x
     : forall X : Set, (X -> X -> Prop) -> Prop
```

If R is well–founded and Φ is a term defining a program on an input $x \in X$ from its values on those $y \in X$ that are R–related to it, then one can define the program `genrec` as:

```
Coq < Variables (X:Set) (R:X->X->Prop).
Coq < Variables (wfR:(well_founded R))
Coq < (Phi:(forall x : X, (forall y : X, R y x -> X) -> X)).
Coq < Definition genrec:=(well_founded_induction wfR (fun _:X=> X) Phi).
genrec is defined
```

where `well_founded_induction` has the following type:

```
Coq < Check well_founded_induction.
forall (A : Set) (R : A -> A -> Prop),
well_founded R -> forall B : A -> Set,
(forall x : A, (forall y : A, R y x -> B y) -> B x) ->
forall a : A, B a
```

now, it is possible to prove the following fixed–point theorem [2]:

```
Coq < Theorem fixpoint: forall (x:X),
Coq < genrec  wfR PhiW x =
Coq < PhiW (fun y:X => fun _:(R y x)=>(genrec  wfR PhiW y)).
```

where the specific Φ is now defined by:

```
Coq < Definition PhiW:forall x:X,(forall y:X,(R y x)->X)->X:=
Coq < fun (x : X) (H : forall y : X, R y x -> X) =>
Coq < match r x with
Coq < | left p => H (f x) (term  p)
Coq < | right _ => x
Coq < end.
PhiW is defined
```

6 Modeling Loops

With the previous fixed–point property, we model the while–loop by the program `while_do` and prove the basic properties that determines it uniquely:

```
Coq < Definition while_do:=fun x:X=>(genrec wfR PhiW x).
while_do is defined
Coq < Lemma while1:forall (x:X), P x -> while_do x = while_do (f x).
Coq < Lemma while2:forall (x:X), ~(P x) -> while_do x = x.
```

In Coq, given `X:Set` and `P:X->Prop`, the inductive type `{x:X | P x}` (in abstract syntax `(sig A P)`) is a `Set`. A term with this type has the form `(exist x p)`

whenever we can find a witness `x:X` and some justification `p:P x`. Such a term proves the constructive existence of some element which satisfies the predicate `P`. The inductive type of the existential quantifier exists also in Coq.

To increase confidence in our model we represent the semantics of the while–loop by proving that $\forall x \in X \cdot \exists n \in \mathbf{N} \cdot \texttt{while_do}\, x = \texttt{itern}\, n\, x$, and extracting from the proof an explicit function expressing n in terms of x.

7 A Simple Example

Our methodology is problem driven. We extract from the real code some components needed to apply our model. Let us take, as an example, the following simple C program:

```
#include <stdio.h>
main()
{  int N;
   printf("enter number:");
   scanf("%d", &N);
     int i=0;
     int sum=0;
     while (i < N ) {
     i=i+1;
     sum=sum+i;}
   printf("%d\n",sum);}
```

with the obvious specification that for all natural n this program calculates the sum of the first n natural numbers.

Our system identifies program variables, initial state, minimal syntax and semantics needed and the loop's body function `f` (the symbol `;` means concatenation):

```
Inductive Var:Set:=
v1:Var|v2:Var.
Inductive aexp:Set:=
V:Var->aexp|Num:Z->aexp|Sum :aexp->aexp->aexp.
Definition s0 (t:Var):=
match t with
 v1 => 0
|v2 => 0
end.
Fixpoint sem (s:States) (a:aexp){struct a}:Z:=
match a with
 V t => (s t)
|Num n => n
|Sum a1 a2 => (sem s a1)+(sem s a2)
end.
Definition f:= (asn d v1 (Sum (V v1) (Num Var 1)))
; (asn d v2 (Sum (V v2)) (V v1)).
```

The constant `d` in section 3 is immediate from the inductive definition of `Var`. Also, we need `P` and `R` in section 4. Note that in this example they are both indexed by the program input n. The actual `P` is extracted to:

```
Definition P (n:Z) (s:States):Prop := (s v1) < n.
```

Although finding relation `R` can be often quite tricky, it is less difficult than finding adequate variant and invariant for a while–loop which, as was already mentioned, is something we do not need in our approach. Here `R` is defined by:

```
Definition R (n:Z) (s1 s2:(States)): Prop:=
(0<= n-s2 v1)/\(n-s1 v1)<(n-s2 v1).
```

Now the user is prompted to prove the three following theorems:

```
Lemma one:forall (n:Z) (x : States), {P n x} + {~ P n x}.
Lemma two:forall (n:Z) (x:States),(P n x)->(R n (f x) x).
Lemma three: forall n:Z,well_founded (R n).
```

which once proved to hold, allow us to define the term `realprog` as the Coq model of our initial C program:

```
Coq < Definition prog:= fun z:Z => (while_do (one z) f (@two z) (three z)).
prog is defined
Coq < Definition realprog:= fun z:Z => prog z s0 v2.
realprog is defined
```

To establish the required specification we use the obvious embedding `nat2z:nat->Z`:

```
Coq < Theorem specification_0:forall z, z<=0->(realprog z)=0.
Coq < Theorem specification_1:forall n:nat,
Coq < (realprog (nat2z (S n)))=(realprog (nat2z  n))+(nat2z (S n)).
```

8 Conclusions and Future Work

A very simple methodology to model imperative programs and their specifications in Coq has been presented. This method does not uses any Hoare–logic tool.

The actual implementation allows, with a friendly user interface guidance, to extract from imperative code the proofs to be built in each case. This not only certifies termination but can also permits to prove constructively the verification of the program specification.

We hope our work, as it can helps to understand semantics of imperative programs, can be useful for both practitioners and students. Our approach can be also seen as a first step for using more sophisticated tools as the mentioned [8].

In order to facilitate the system use, more automatic behavior must be added to the interface. Also to enrich the programming language more program constructs ought to be modeled in the future.

References

1. Aczel, P.: Introduction to Inductive definitions. In: Barwise, J. (ed.) Handbook of Mathematical Logic, North Holland (1997)
2. Balaa, A.: Fonctions récursives générales dans le calcul des constructions. PhD. Theése. Université de Nice–Sophia Antipolis (2002)
3. Bertot, I., Castéran, P.: Interactive Theorem Proving and Program Development. Springer, Heidelberg (2004)
4. Bertot, I., Capretta, V., Barman, K.: Type–Theoretic Functional Semantics. In: Carreño, V.A., Muñoz, C.A., Tahar, S. (eds.) TPHOLs 2002. LNCS, vol. 2410, pp. 83–97. Springer, Heidelberg (2002)
5. Bove, A.: Simple General Recursion in Type Theory. Technical Report. Chalmers University of Technology, Goteborg (2000)
6. Coquand, T.: An Introduction to Type Theory. Notes of the FPCL summer school, Glasgow (1989)
7. Coquand, T., Pauling-Mohring, C.: Inductively defined types. In: Martin-Löf, P., Mints, G. (eds.) COLOG-88. LNCS, vol. 417, Springer, Heidelberg (1990)
8. Filliâtre, J.-C.: Why: a multi-language multi-prover verification tool, Research Report, LRI, 1366 (March 2003)
9. Barthe, G., Forest, J., Pichardie, D., Rusu, V.: Defining and reasoning about recursive functions: a practical tool for the Coq proof assistant
10. Nipkow, T.: Winskell is (almost) right: Towards a mechanised semantics test–book. In: Chandru, V., Vinay, V. (eds.) Foundations of Software Technology and Theoretical Computer Science. LNCS, vol. 1180, Springer, Heidelberg (1996)
11. Pauling-Mohring, C.: Inductive Definitions in the System Coq–Rules and Properties. In: Bezem, M., Groote, J.F. (eds.) TLCA 1993. LNCS, vol. 664, Springer, Heidelberg (1993)
12. Nordström, B.: Terminating General Recursion. BIT, vol. 28 (1988)
13. Pfenning, F., Pauling-Mohring, C.: Inductively defined types in the Calculus of Constructions. In: Schmidt, D.A., Main, M.G., Melton, A.C., Mislove, M.W. (eds.) Mathematical Foundations of Programming Semantics. LNCS, vol. 442, Springer, Heidelberg (1990)
14. Werner, B., Paulin Mohrin, C.: ENSTA: Course Notes (1995)
15. Winskell, G.: The Formal Semantics of Programming Languages, an introduction. Foundatios of computing. The MIT Press, Cambridge (1993)

Using Coq to Understand Nested Datatypes*

A. Blanco, J.E. Freire, and J.L. Freire

University of A Coruña, Spain
{blanco, efreire, freire}@udc.es
University of A Coruña, LFCIA, Dept. of Computer Science, Faculty of Informatics, 15071 A Coruña, Spain

Abstract. Nested datatypes can be defined in a Hindley-Milner system but it is difficult to implement functions on them. This paper analyzes the requirements that must be satisfied in order to implement programs (catamorphisms and anamorphisms) on these types. A solution, using Hagino's approach, is carried out here both in Haskell, using rank 2 signatures, and in the Coq proof assistant system where we have polymorphic recursion and also the capability to prove the correspondent programs specifications.

1 Introduction

Algebraic data types can be either:

- *regular types*, that can be inductive (natural, lists, trees,...) or coinductive (streams, naturals with infinite,...) and come through continuous and co-continous functors; or else
- *nested types* (not regular or heterogeneous) like, for example, the type in Haskell `data Nest a =NinN | ConsN(a,Nest(a,a))`, where a recursive call is made with different arguments than the original one.

This paper analyzes the requirements that must be satisfied in order to implement programs (catamorphisms [2] and anamorphisms [14]) over nested types. A solution, using Hagino's approach [7], is carried out here, both in Haskell [8], using rank 2 signatures, as well as in the Coq [1] proof assistant system where we have polymorphic recursion and also the capability to prove the correspondent programs specifications.

In the following, we solve some particular cases and, through them, we propose the basic requirements to define such a morphisms. Afterwards, we offer a global methodology for the general case. For that, we provide a new categorical characterization of nested types.

Let us consider first the example of the nested type rope implemented in Coq:

```
Inductive rope:Set->Set->Type :=
  Rnil:forall A B:Set,rope A B
  |Rcons:forall A B:Set,A->rope B A->rope A B.
```

where the constructor `Rcons` has inverted its arguments in the recursive call.

* Supported by: MEC, Spain, Project TIN2005-08986. FARMHANDS.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 210–216, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Using our method we can define the catamorphism `fold` for this type using structural recursion in Coq:

```
Fixpoint fold (A C X Y:Set) (theta:A->C->C)(c0:C)
(rs:rope X Y) {struct rs}: (X->A)->(Y->A)->C:=
match rs in (rope X0 Y0) return ((X0->A)->(Y0->A)->C) with
  Rnil X1 Y1->fun f:X1->A->fun g:Y1->A->c0
  |(Rcons X1 Y1 a rs')->fun f:X1->A->fun g:Y1->A->
    theta (f a) (fold C Y1 X1 theta c0 rs' g f)
end.
```

The bold part shows clearly where we are using polymorphic recursion.

The use of nested types allows a more exact implementation of types with regard to their specifications, relieving the programmer of a series of expendable tasks.

Some important examples of using nested types are shown next:

- Okasaki implements square matrices in the `fast exponentiation` application [12];
- Bird and Paterson model Bruijn's λ-calculus [3];
- Hinze describes numeric representations with high order nested types (where the parameter passed to the data type is the type itself) [9].

Recently the question of implementing these nested types and testing their properties within the context of theorem provers has emerged [11], [13]. In this work we follow that line from a different perspective. The paper is structured as follows. Section 2 provides background information for the paper. Our definition of nested type, based on Hagino's approach, and the fold and unfold constructions in Haskell are given in section 3. Finally, in section 4 we present an example to illustrate how this approximation goes in Coq where the specifications can be verified.

2 Non-regular Types

These types have begun to be analysed in the last few years [2], [3], [4], [5], [6].

One problem that came up when studying these types was the construction of morphisms determined by their initial or final universal condition. For example, given the functor F, representing the abstract structure of a regular type, the function

$$mapF : (A \to B) \to (F\ A) \to (F\ B)$$

is an example of the corresponding catamorphism

$$foldF\ \xi_B\ f,$$

where $\xi_B : (F\ B) \to B$ is the structure associated to the type $(F\ B)$ and f is a morphism between A and B.

In the case of nested types [4], [5] this instantiation is not so immediate. Several attempts have been made to solve this problem. Bird & Meertens [3] and Blampied [5] give a solution using local universal quantification but (as they themselves point out) it is only a partial solution as it generates instantiation problems. Taking the problem up again, Bird & Meertens [4] generalize the catamorphism and referred to as `generalised fold`.

In [10] an efficient generalized fold ($efold$) is constructed leaving out the *map* required in Bird's original version. As an auxiliary tool, he adds an additional argument and continues using polymorphic arguments:
$efold :: (\forall a.n\ a) \rightarrow (\forall a.m\ a \rightarrow n(Pair\ a) \rightarrow n\ a) \rightarrow (\forall a.Pair(m\ a) \rightarrow m(Pair\ a)) \rightarrow (l\ a \rightarrow m\ b) \rightarrow Nest(m\ a) \rightarrow n\ a$

$$efold\ e\ f\ g\ h\ NilN\ =\ e;$$

$$efold\ e\ f\ g\ h\ (ConsNx\ xs)\ =\ f\ x\ (efold\ e\ f\ g\ (g.pair\ h)\ xs)$$

In order to achieve implementations, functor categories as well as natural transformations are used in all of these mentioned approaches. In addition, only folds are analyzed, leaving out unfolds (where the nested type plays the role of a terminal object) and hylomorphisms.

3 Non-regular Data Types

Let T be a category. Given functors $F,\ G : \mathrm{T} \times \mathrm{T} \rightarrow \mathrm{T}$ and one non identity endofunctor $H : \mathrm{T} \rightarrow \mathrm{T}$, a functor K determines a nested type if, for every type A there exist an isomorphism $in_A : F(A, K(H(A))) \simeq G(A, K(A))$.

Let K be such a functor. Then, given $\xi : F(A, H(B)) \rightarrow G(A, B)$ and a morphism $p : C \rightarrow A$, to ensure the existence of $ExtK_{A,C}\ \xi\ p : K(C) \rightarrow B$ satisfying the dialgebras morphism condition [7]:

$$in_C; G(p, ExtK_{A,C}\ \xi\ p) = F(p, ExtK_{H(A),H(C)}\ \rho(\xi)\ \tau(p)); \xi \qquad (1)$$

We need a new function

$$ExtK_{H(A),H(C)}\ \rho(\xi)\ \tau(p) : K(H(C)) \rightarrow H(B)$$

which in turn ought to come from another functions

$$\rho(\xi) : F(H(A), H2(B)) \rightarrow G(H(A), H(B))$$

and

$$\tau(p) : H(C) \rightarrow H(A).$$

The latter, as H is a functor, becomes simply $H(p)$. The problem raises when we want to obtain the former.

Unlike to what is done in [3] and [10], our proposal does not require leaving the underlying category T.

As it can be observed in the equation corresponding to the functional $extK$ (1), we should restrict the constructors types to those which allow to obtain the required $\rho(\xi)$ from given ξ. This can be done using classes in Haskell [8].

Let see how to use Haskell classes to implement the generic fold or catamorphism. Due to space limitations we will use the `Llist` type as an example to illustrate our approach.

```
data Llist a= NilL
| ConsL(a,Llist [a])
```

Values of this type are lists whose arity increases with successive applications of `ConsL`. In this case we have functors $F(A,B) = () + A \times B,\ G(A,B) = B,\ H = List$, where $Llist$ should verify $() + A \times Llist(ListA) \simeq Llist\ A$.

If the category used is predistributive ω-complete and ω-cocomplete, we have:

$$Llist(A) \simeq () + A \times Llist(List(A)) \simeq$$

$$() + A \times (() + List(A) \times Llist(List(List(A)))) \simeq$$

$$() + A \times (() + List(A) \times (() + List(list(A)) \times Llist(List(List(List(A)))))) \simeq$$

$$() + A + A \times List(A) + A \times List(A) \times List(List(A)) + \ldots$$

$$() + A + A \times List(A) + A \times List(A) \times List^2(A)) + \ldots \simeq \coprod_{n \in nat} \prod_{n \in nat} List^n(A),$$

with $H0(A) = ()$.

Therefore, we can state the existence of type $Llist(A)$. Furthermore, if $G(A,B) = B$ and $F(A,B) = A + B$ or $F(A,B) = A \times B$, it follows that functor K also exists.

It is observed that for the construction of $\rho(\xi)$, it is necessary to have two morphisms:

- one that from an element in C returns one in $[C]$,
- and another that from a given function $B \times [C] \to C$ makes possible the construction of one with type $[B] \times [[C]] \to [C]$.

These two conditions can be established in Haskell as follows:

```
class Valid f where
    fValid1 :: (f x)->f [x]
    fValid2 :: ((x,f[x])->f x)->([x],f [[x]])->f[x]
```

Therefore, such transitions must be provided in each specific case to obtain the catamorphism `extLlist`:

```
extLlist :: Valid f => (f a)->((a,f [a])->f a)->(c->a)->(Llist c)->f a
extLlist m e h NilL = m
extLlist m e h (ConsL(x,xs)) =
e(h x,extLlist (fValid1 m) (fValid2 e) (map h) xs)
```

The construction of the anamorphism `anaLlist` over `Llist` is as follows:

```
class AnaValid f where
    anaValid1 :: (f x->Bool)->f [x]-> Bool
    anaValid2 :: (f a->(a,f [a]))->f [a]->([a],f[[a]])

anaLlist :: AnaValid f =>
(f a->Bool)->(f a->(a,f [a]))->(a->b)-> f a->Llist b
anaLlist p th h y = if p y then NilL else
ConsL( h x,anaLlist (anaValid1 p) (anaValid2 th) (map h) y')
where (x,y')=th y
```

Then, we can also define their composition. This turns out to be the corresponding hylomorphism `hylLlist`.

```
hylLlist :: (Valid f, AnaValid g) =>
(g a->Bool)->(g a->(a,g [a]))->(a->b)->
(f c)->((c,f [c])->f c)->(b->c)->(g a)->f c

hylLlist p th h m e k y= (foldLlist m e k) (anaLlist p th h y)
```

and optimizing by applying deforestation:

```
opthylLlist :: (Valid f, AnaValid g) =>
(g c -> Bool) -> (g c -> (c,g [c])) -> (c -> c) -> f c ->
((c,f [c]) -> f c) -> (c -> c) -> g c -> f c

opthylLlist p th h m e k y=
   if p y then m
     else
      e(k(h x),foldLlist (fValid1 m) (fValid2 e)
      (map h) (anaLlist (anaValid1 p) (anaValid2 th) (map h) y'))
      where (x,y')=th y
```

This methodology gives good results even with higher order nested types.

4 Interpretation of Nested Types in Coq

Due to space limitations we will focus here only on the case of Bruijn's notation for λ–calculus. This type is defined by the fixed–point equation:

$$\mathbf{Term\ A} \simeq A + \mathbf{Term\ A} \times \mathbf{Term\ A} + \mathbf{Term}\ (\mathbf{Incr\ A})$$

where $Incr(A) \simeq (\,) + A$.

Using Coq we have:

```
Inductive Incr  (V:Type) : Type :=
|Zero : Incr V
|Succ : V->Incr V.
```

```
Inductive  Term : Type|Type :=
|Var : forall (V:Type), V->Term V
|App : forall (V:Type), Term V->Term V->Term V
|Lam : forall (V:Type), Term (Incr V)->Term V.

Structure FoldT:Type :=
f:Type -> Type;
first: forall A:Type,Incr A->f (Incr A);
second: forall A:Type,(f A->f A->f A)->(f (Incr A)->f (Incr A)->
f (Incr A));
third:forall A:Type,(f (Incr A)->f A)->(f (Incr (Incr A))->f (Incr A)).

Definition H1 : FoldT -> (Type -> Type).

Fixpoint extT (B C:Type)(f:FoldT)(f1:B->((H1 f) B))
(f2: (H1 f) B->(H1 f) B->(H1 f) B)
(f3:((H1 f) (Incr B))->(H1 f) B) (ts:Term C){struct ts}: (C -> B) ->
(H1 f) B
:= match ts in (Term C0) return ((C0->B)->((H1 f) B)) with
  |Var C1 v => fun h:C1->B => (f1 (h v))
  |App C1 ts' ts'' => fun h:C1->B =>
          f2 (extT B C1 f f1  f2 f3 ts' h) (extT B C1 f f1  f2 f3 ts'' h)
  |Lam C1 ts' => fun h:C1->B =>
          f3 (extT (Incr B) (Incr C1) f  (first f B  ) (second  f  B f2)
          (third f B f3) ts' (mapI C1 B h))
end.
```

The Coq 8.0 code of the correspondent proofs of properties for this (and also others) implementation can be found at `www.dc.fi.udc.es/staff/blanco`.

References

1. Bertot, I., Castéran, P.: Interactive Theorem Proving and Program Development. Springer, Heidelberg (2004)
2. Bird, R., Meertens, L.: Nested datatypes. Mathematical foundations for computing. LNCS (1998)
3. Bird, R., Paterson, R.: de Bruijn notation as a nested datatype. Journal of Functional Programming 9(1) (1999)
4. Bird, R., Paterson, R.: Generalised folds for nested datatypes. Formal Aspects of Computing 11 (1999)
5. Blampied, P.A.: Structural Recursion for non-Uniform Data Types. Phd. Thesis. University of Nottingham (2000)
6. Martin, C., Gibbons, J.: On the Semantics of Nested Datatypes. Information Processing Letters 80 (2001)
7. Hagino, T.: A categorical programming language, PhD. Thesis. Edinburgh University (1987)
8. Hall, C.V., Hammond, K., Peyton Jones, S.L., Wadler, P.L.: Type classes in Haskell. ACM Transactions on Programning Languages and Systems (1996)
9. Hinze, R.: Numerical Representations as Higher–Order Nested Datatypes (1998), `http://citeseer.ist.psu.edu/hinze98numerical.html0`

10. Hinze, R.: Efficient Generalized Folds. In: Proceedinga of the Second Workshop on Generic Programming, WGP 2000 (2000)
11. Matthes, R.: Verification of programs on truly nested datatypes in intensional type theory. Mathematically Structured Functional Programming (2006)
12. Okasaki, C.: From Fast Exponentiation to Square Matrices: an Adventure in Types. ACM 1-58113-111-9/99/0009 (1999)
13. Rodriguez, D.: Verification of Iteration and Coiteration Schemes for Nested Datatypes in Coq. Institut für Informatik der Ludwig–Maximilians–Universität, München. Technical Report (2006)
14. Vene, V.: Categorical Programming with Inductive and Coinductive Types. Dissertationes Mathematicae Universitatis Tartuensis, 23 (2000)

Towards a Decentralized and Structured Network of P2P Public Information Screens*

Carlos Abalde, Víctor M. Gulías, and Laura M. Castro

MADS Group. Computer Science Department. University of A Coruña, Spain
{cabalde, gulias, lcastro}@udc.es

Abstract. This paper presents our results designing a peer-to-peer (P2P) video content distribution network (CDN). Based on a distributed hash table (DHT) algorithm and developed as an extension of an existing on-demand streaming server, this CDN is designed to improve server streaming capacity without any infrastructure upgrade to deal with large networks of information screens. Experimental results show that the design is an appropriate approach for video content distribution with interesting scalable and availability features.

Keywords: P2P, Content Distribution Network, Distributed Hash Table.

1 Introduction

Flexible, scalable and fault tolerant content distribution networks (CDNs) are demanded in order to publish huge amount of information which is going to be accessed by a large number of users. Two general approaches are idetified to do it [1]: *infrastructure-based content distribution*, improving server infrastructure in a client/server framework, and *peer-to-peer content distribution*, replacing the client/server model by a *peer-to-peer* (P2P) approach. Intuitively, a P2P approach is better suited to deal with mass-scale content distribution. However, this can only be achieved with more effort in terms of coordination, resource management, heterogeneity...

In this research, we deal with multimedia content distribution (specifically, video contents). A video content service has large media size contents, read-only sequential access, high service time and low lookup latency. Our goal is to design a video CDN suited to the scenario discussed in section 2, combining both infrastructure-based (an existing streaming server, VoDKA [2]) and P2P (the main practical contribution of this research) approaches.

The paper is structured as follows. First, our working scenario is presented in section 2. Then, some background knowledge about P2P architectures and content location protocols based on distributed hash tables is introduced in section 3. Section 4 discusses our proposed solution, and section 5 presents some performance evaluation results. Finally, we present our conclusions.

* Partially supported by MEC project TIN2005-08986 and Xunta de Galicia project PGIDIT06PXIC105164PN.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 217–224, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Problem Description

The target scenario is a network of public *video-information screens* scattered among several *locations* (figure 1). At each location, there may be several video-information screens LAN-connected. Media is primarily stored in a central multimedia server (a VoDKA system), WAN-connected with each location (typically, xDSL link). Each video-information screen has a *programme*, a grid of *media objects* (MOs) that defines the particular media scheduling, and *next programmes* which will be valid in the near future (usually published on a daily-base).

In principle, each screen must request to the central server all the scheduled MOs *before* they are required for playback. The deadline is a *timestamp*, a combination of the programme start and the actual first appearance of the MO in the programme. In order to avoid fetching MOs every time, each screen includes a local cache storage.

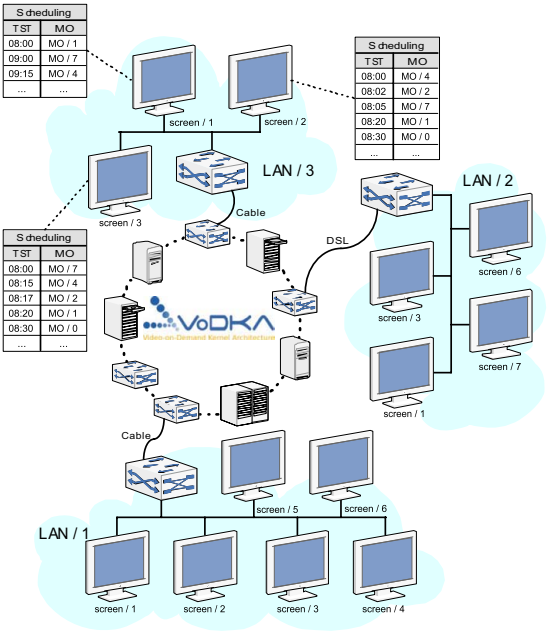


Fig. 1. Content distribution scenario

This is a naïve approach, as long as several screens in the same LAN may request the same MO (same MO is sent through WAN link several times to the same location), which is a quite common case because, in practice, most of the programmes share a great deal of content. Therefore, in order to improve this situation, screens at the same location should *cooperate* to avoid unnecessary traffic with central storage, which, as an isolated component, is a complete infrastructure-based CDN.

This intra-LAN cooperation must fulfill some important requirements: (1) it minimizes coupling, allowing screens to operate completely or partially isolated in -probably temporal- adverse conditions; (2) it avoids single points of failure inside each LAN, in order to reduce management complexity and improve availability; (3) it is designed as an scalable architecture at LAN level, (4) it allows efficient screen departure and arrivals; (5) it keeps the server as a central storage and administrative point; and (6) it minimizes deployment and maintenance costs.

2.1 Solution Sketch

The problem shown above suggests the definition of a solution that exploits MO sharing and temporal locality among all screen schedulings and, in particular, screens from the same location. Therefore, a coordination scheme for every location is required to improve caching effort while fulfilling the problem requirements. The benefits would be directly proportional to the correlation among screen schedulings. In the ideal case, when all the screens in the same LAN have identical schedulings, each location can be considered equivalent to one single screen, with independence from the number of screens at that location.

The obvious solution of defining proxy nodes to coordinate LAN-access to central server is inadequate because it introduces a single point of failure and it does not scale. A refinement of the former idea is to introduce some supernodes at each location in charge of indexing and manage location global state, basically what is being loaded or has already being cached by which screen. This solution would complicate management, introduce scalability constraints and, again, expose new points of failure in the local network (the supernodes themselves). Next section presents some backgroud information related with our final approach.

3 P2P Content Distribution

Any P2P content distribution system relies on a network of peer computers and connections among them. A survey of P2P material can be found at [3]. The topology, structure and degree of centralization of the overlay network, and the routing and location mechanisms it employs for messages and content look-up, are crucial to the operation of the system. They affect its fault tolerance, self-maintainability, adaptability to failures, performance, scalability, and security.

P2P content distribution architectures can be classified according to their centralization degree and internal structure. Here we are interested on *fully decentralized structured architectures* since they are the best scalable design and the most suitable approach for our CDN design. Two typical problems are identified in these architectures: (a) *Overlay network creation and maintenance*, how to build peers structure without any centralized servers or supernodes, and how to keep this mechanism scalable and fault tolerant; (b) *Look-up*, how to find an item in a P2P system in a scalable way, without any centralized servers [4]. Both questions are addressed using *distributed hash tables* (DHTs). In a DHT, data

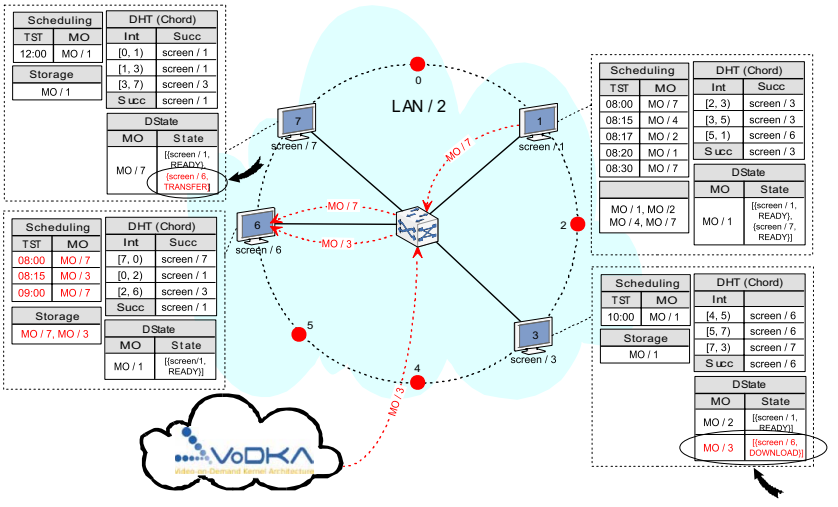


Fig. 2. Coordination using Chord-based DHT example

items are inserted and looked up using a unique key. In order to implement a DHT, the underlying algorithm must be able to determine the node responsible for storing the data associated with a given key. Hence, each node handles information (e.g., the IP address) of a small number of other *neighbor* nodes, defining an overlay network, and routes messages to store and retrieve keys [4].

Up-to-date DHT algorithms stress the ability to scale well to large numbers of nodes, to locate keys with low latency, to handle node arrivals and departures scalably, to ease the maintenance of per-node routing tables, and to balance the distribution of keys evenly among the participating nodes. Chord [5], Pastry [6], or Koorde [7] are some popular DHT algorithms. Each one suggests a specific overlay network where all the nodes are logically interconnected. Different overlay networks have different advantages and drawbacks in terms of look-up latency, resilience to node failures, etc.

4 The Coordination Algorithm

We propose a design built on top of an structured and fully decentralized P2P architecture based on a DHT. A DHT is defined at each location (local network) to store screens state, using the MO identifier as key and storing information about MO state in the local network (whether it is being loaded in a node from central server or nodes store a copy locally). Therefore, every screen in the local network has a (mostly accurate) global vision of location state to implement the coordination strategy.

Figure 2 shows an overlay network on top of a 4-node LAN (`screen/1`, `screen/3`, `screen/6` and `screen/7`). Chord has been chosen as DHT algorithm,

defining a logical ring that tights nodes (screens) and keys (MO identifiers). For each screen, the figure shows its MO scheduling (combining current and future programmes), the available MOs in its local cache, the routing table (*finger table*, using Chord's terms) for the overlay network and the key/value pairs stored in the DHT. The state and location of all copies are distributed (and replicated, optionally) and it can be efficiently retrieved by each screen ($\mathcal{O}(logN)$, being N the maximun number of screens in the local network). In the example, `screen/6` starts with no scheduling and and no MO in cache. As soon as its scheduling is updated, the coordination algorithm is triggered to transfer to its local cache `MO/3` (from central storage server) and `MO/7` (locally transfered from `screen/1`), updating the DHT information properly.

4.1 Formalization of the Scenario

Let be $L = \{L_1, L_2, \ldots\}$ a set of LANs, where each L_i represents screen nodes $L_i = \{L_i^1, L_i^2, \ldots\}$. Each screen L_i^j in LAN L_i receives and locally stores programmes (timed sequence of media to play):

$$sch_v(L_i^j) = \{(MO_1, tst_1), (MO_2, tst_2), \ldots\}$$

where $v \in \mathbb{N}$ represents programme version (current and future programmes), MO is the identifier of the media object and tst is the timestamp when the MO must be available. Overall scheduling for L_i^j is:

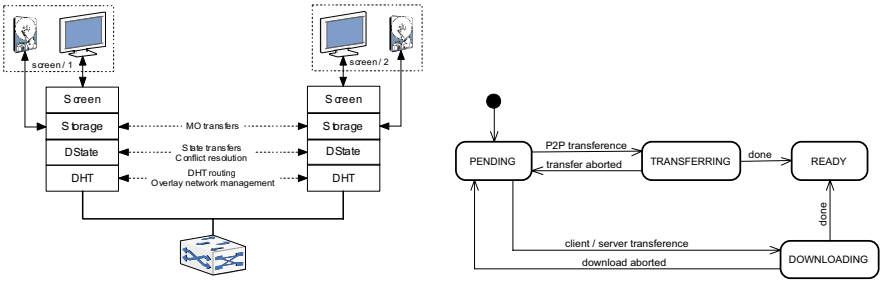
$$sch(L_i^j) = sch_v(L_i^j) \otimes sch_{v-1}(L_i^j) \otimes \ldots \otimes sch_0(L_i^j)$$

where $\otimes$ is defined as:

$$\begin{aligned} A \otimes B = {} & \{(MO, t_1) | (MO, t_1) \in A \wedge (MO, t_2) \notin B\} \cup \\ & \{(MO, t_2) | (MO, t_1) \notin A \wedge (MO, t_2) \in B\} \cup \\ & \{(MO, min(t_1, t_2)) | (MO, t_1) \in A \wedge (MO, t_2) \in B\} \end{aligned}$$

Node L_i^j has a layered architecture (figure 3(a)): (1) the *screen layer* is the frontend interface in charge of recomputing $sch(L_i^j)$ and notifying it to the lower layer everytime a new programme version, $sch_{v+1}(L_i^j)$, is received; (2) the *storage layer* takes care, knowing what and when MOs are going to be required, of applying the coordination algorithm using local and distributed state (handled by DState layer); (3) the *DState layer* stores (part of) the global state, performs state transfers among nodes and solves conflicts due to asynchronous behavior of nodes; (4) the *DHT layer* is in charge of defining the overlay network and updating its routing information.

Global state is distributed using the DHT, indexed by MO identifier. The value stored for each MO index is the list of available copies at the local network and, for each copy, the node identifier and MO's current state according with the state diagram shown in figure 3(b): (1) PENDING: the MO is being required but not decided yet how to transfer it to local cache. It is both the initial state and the state of aborted transferences because of external reasons. (2) DOWNLOADING:



(a) Node architecture (b) MO state diagram

Fig. 3. Node behaviour

MO is being transferred from central storage. (3) TRANSFERRING: MO is being transferred from another screen in the same local network. (4) READY: MO is locally available.

Algorithm 1. Coordination Algorithm

```
if stg(MO) ∈ {DOWNLOADING, TRANSFERRING} then
  vodka.notify(MO, tst)
else if stg(MO) ≡ PENDING then
  if copies = dstate(MO) then
    vodka.notify(MO, tst)
    node = select_peer(copies)
    node.transfer(MO))
    stg(MO) = dstate(MO) =
      {TRANSFERRING, node}
  else
    vodka.download(MO, tst)
    stg(MO) = dstate(MO) = DOWNLOADING
  end if
end if
```

At the storage layer, three events are identified that require the use of coordination algorithm to minimize the interaction with central storage as long as possible: (a) new PENDING MO appears (b) MO deadline update (c) MO transfer cancellation (in both TRANSFERRING or DOWNLOADING state). In this algorithm, $dstate(MO)$ offers access to MO state using DHT layer and $select_peer(copies)$ chooses one of the possible peers to perform MO transference (from a READY MO copy or, if not possible, from the cheaper TRANSFERRING or DOWNLOADING copy available). Finally, $stg(MO)$ accesses to local copy.

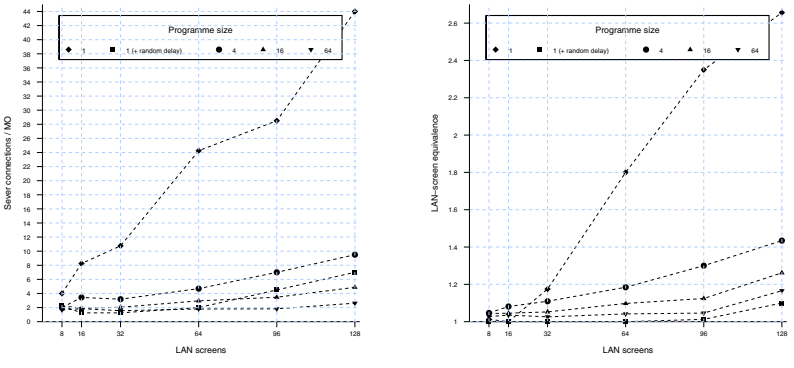
Due to asynchronous behavior, a wrong decision that affects consistency may happen. This must be detected and solved by the DState layer. An example is the situation with several screens downloading the same MO from central storage.

The election of the leader node in charge of solving the conflict is performed thanks to DHT MO/node mapping, establishing a dialog with involved nodes to abort unnecessary transferences. Then, coordination algorithm is re-executed.

5 Prototype Benchmarking

A prototype implementation has been developed using the concurrent functional language Erlang [8] and Chord DHT algorithm. Each screen node is running on different Erlang virtual machines (EVM), and several EVM are deployed on different computers. An additional EVM is in charge of gathering monitoring information from all the screen nodes.

In the experiment, we consider one LAN with a variable number of screens, ranging from 8 to 128, deployed on 4 physical computers. Simulation starts with no scheduling and no media on local caches and an stable overlay network. After that, four bursts of workload are generated with new programme versions including a set of fresh MOs, exactly the same for each screen, distributed simultaneously. The size of the programme is what we consider as concurrency level, which range in the experiment from 1 to 64.



(a) Server connections / MO (b) LAN-screen equivalence

Fig. 4. Some prototype benchmarks

Figure 4(a) shows the number of connections to central server for each different MO, assuming that there is no partitioning in the DHT. With higher concurrency levels, the results approximate to ideal value. For example, with 128 peers and concurrency level 64, server receives 2.62 connections per MO (ideal case 1, worst case 128). As can be observed, system behavior is not affected significatively by high concurrency or large screen populations. Surprisingly, with low concurrency (from 1 to 4 new inserted MOs) and many peers, the average number of connections per MO drops significatively (i.e., with 128 peers and 1 MO, average drops to 44.00, most of them conflicts that are immediately cancelled). The cause of this pathological behavior is that, with only a few MOs updated

simultaneously on several peers, there is a high possibility of collisions for the first scheduled media. In order to prevent this behavior it is enough to introduce a small random delay at the very beginning of the coordination.

Since 1 is the ideal number of connections per MO, any additional connection is a conflict that must be cancelled. It is interesting to measure the impact of this situation on central storage bandwidth consumption. Figure 4(b) shows that the whole local network is similar to 1.2 screen nodes working with no coordination at all. As in the previous measurement, increasing the number of peers does not affect significatively bandwidth usage. Again, results are far from ideal value because the pathological problem already commented, that can be easily avoided.

6 Conclusions and Future Work

We presented a design for a P2P video CDN based on a direct application of a DHT algorithm to extend an existing on-demand streaming server. As shown in the experimentation, our strategy solves the problem with no additional extension of the central architecture, with interesting scalable and availability features. Though we have ommitted the actual integration with the streaming server in the sake of space, it must be upgraded to manage central bandwith. Our following step is to evolve our CDN design towards an auto-organized and decentralized P2P resource management.

References

1. Padmanabhan, V., Wang, H., Chou, P., Sripanidkulchai, K.: Distributing streaming media content using cooperative networking (2002)
2. Gulías, V.M., Barreiro, M., Freire, J.L.: VoDKA: Developing a video-on-demand server using distributed functional programming. Journal of Functional Programming 15(4) (2005)
3. Androutsellis-Theotokis, S., Spinellis, D.: A survey of p2p content distribution technologies. ACM Computing Surveys 36(4), 335–371 (2004)
4. Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R., Stoica, I.: Looking up data in P2P systems. Commun. ACM 46(2), 43–48 (2003)
5. Stoica, I., Morris, R., Karger, D., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable P2P lookup service for internet applications. Technical Report TR-819, MIT (March 2001)
6. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale P2P systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218, pp. 329–339. Springer, Heidelberg (2001)
7. Kaashoek, M.F., Karger, D.R.: Koorde: A simple degree-optimal distributed hash table. In: Proceedings of the 2nd International Workshop on P2P Systems (2003)
8. Armstrong, J.L., Williams, M., Virding, R., Wilkstrom, C.: Erlang for Concurrent Programming. Prentice-Hall, Upper Saddle River, NJ, USA (1993)

NowOnWeb: News Search and Summarization

Javier Parapar, José M. Casanova, and Álvaro Barreiro

IRLab, Department of Computer Science , University of A Coruña,
Campus de Elviña s/n, 15071, A Coruña, Spain
{javierparapar,jcasanova,barreiro}@udc.es
http://www.dc.fi.udc.es/irlab

Abstract. Agile access to the huge amount of information published by the thousands of news sites available on-line leads to the application of Information Retrieval techniques to this problem. The aim of this paper is to present *NowOnWeb*, a news retrieval system that obtains the articles from different on-line sources providing news searching and browsing. The main points solved during the development of *NowOnWeb* were: article recognition and extraction, redundancy detection and text summarization. For these points we provided effective solutions that put all them together had risen to a system that satisfies, in a reasonable way, the daily information needs of the user.

1 Introduction

Web news has become, in the last years, one of the most used services in Internet. The users have changed the way of access to the information, preferring the web news over other traditional formats. All these joined to the amazing growing of the number of news sites available on Internet produced the overwhelm of the manual techniques to cover all this information. In this situation the use of Information Retrieval (IR) strategies has been a successful solution.

The application of IR techniques solves some of the main problems that rise from the user experience, i.e., the wasting of time in browsing the dozens of different interesting sites for an user, the high level of news redundancy across the multiple sources, and the necessity of searching for specific topics. These problems were previously approached in the IR field, the adaptation of the applied solutions to the *news problem* and the creation of new custom techniques resulted in an specific research field called News-IR.

In this context we present a News-IR system to manage the articles from diverse on-line sources and capable to provide search ability, redundancy detection and summarization. We called it *NowOnWeb*[1], it is based on our previous research and solutions in the IR field also with the help of open source IR products and technologies that were customized in order to integrate together a suitable solution to deal with the user requisites.

Within this paper our aim is to introduce *NowOnWeb* main characteristics, the technologies used, and the key points of the system architecture. Also we will

[1] An operational version is available in http://nowonweb.dc.fi.udc.es

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 225–232, 2007.
© Springer-Verlag Berlin Heidelberg 2007

break down the three main research topics for which we provided useful solutions, i.e., news recognition and extraction, redundancy detection and summary generation. We also present an overview of the previous related work in the news area to illustrate the advantages and innovations of our product, and our future work lines.

2 System

NowOnWeb has result in a complete News-IR system structured in a Model-View-Controller (MVC) Java EE web application that compress the crawling, recognition, extraction, searching, browsing, and summarization of web news.

Characteristics. As our system tries to solve the user requirements, it offers a certain number of characteristics result of those demands. The main system capabilities are:

- Configurable temporal event based crawling of the defined source journals.
- Incremental indexing policy with customizable temporal window, i.e, the time interval from where the articles that are showed to the users in the search results are extracted.
- Flexible and efficient news article recognition and extraction algorithm that makes possible the dynamic adding of new news sources just defining its URL.
- The system provides news searching through user defined queries about an specific topic among all the sources with available articles.
- Adapted redundancy detection strategy to the news ambit, that produces groups of related news that answer to the user query. This frees a lot of user time wasted in reading repeated news.
- A summarization techniques based on sentence retrieval and novelty detection. News summaries are showed to the user instead of the whole articles, reducing again the time needed for the user to be up-to-date.
- Generated summaries are query centered in the sense that are generated based on the information requirements expressed by the user question. Also the user can ask for any summary of any of the indexed articles.
- The application produces a multiple format output, The results can be displayed directly in the web page, in Portable Document Format (PDF) as a news report, or through RSS syndication systems.
- *NowOnWeb* offers to the registered users personalization options, like feeds or source selection, i.e, an user can choose the set of sources that will to contribute to his search results.

Architecture. As we exposed before, the system is structured in a MVC web-application with a component-based design. The design by components allowed us to develop independently each component.

The exposed design isolates the solution to each one of the NewsIR issues in the different components. It makes easier the system maintenance and also

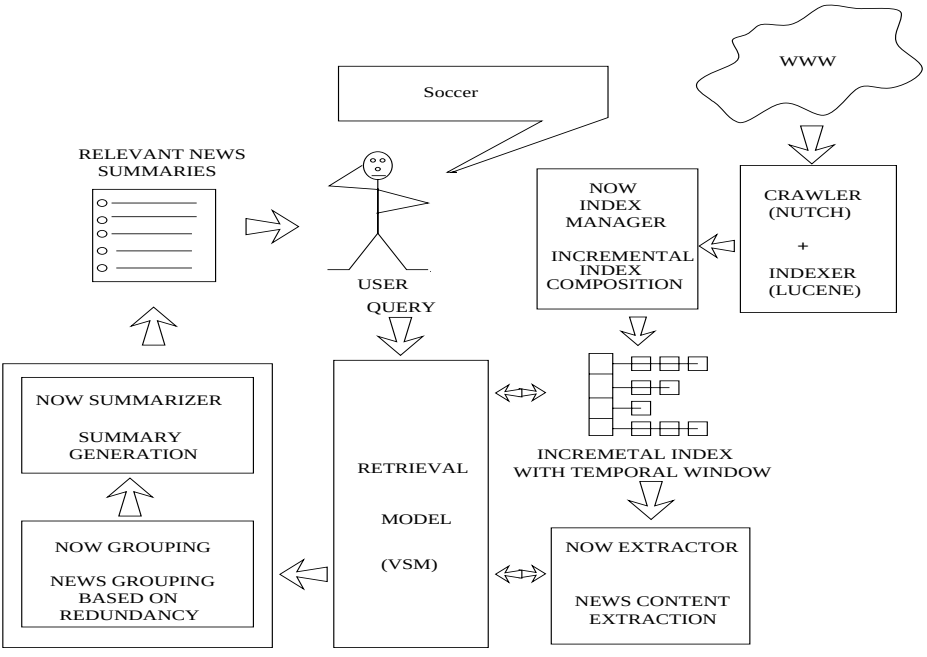


Fig. 1. System componet-based architecture overview

the optimization and addition of new algorithms and solutions. Figure 1 is a general overview of our system architecture, it shows the main components of the application and how they interact in the process to elaborate the answer to an user query.

The responsibility of each component is clearly defined and isolated in the design :

1. The first component, in the logical order of the process, comprises the crawler and the indexer. The crawler is in charge of obtaining the web-pages from the defined news websites, for this task we use a customization of Nutch [2]. After the crawling process an index is generated, the indexing technology in *NowOnWeb* is based on Lucene [4].
2. The next step into the workflow implies the implementation of the temporal incremental index. It will maintain the working index updated and comprising the articles indexed.
3. After crawling from a temporal window a phase of content extraction is done. The recognition and extraction of articles is carried out by the extractor component. This module will differentiate between articles and non-articles pages and, from those that have an article inside, will extract the different news fields like title, image, body, etc.

4. All the documents are indexed and searched based on the Vector Space Model (VSM) [9]. The VSM is an algebraic model used in information retrieval where the documents are represented as vectors of index terms.
5. The retrieval model gives a ranking of relevant documents for an user query. The phase of redundancy detection groups the documents strongly related and redundant together. This is the task of the news grouping component.
6. Once the redundancy groups are created, for each of them the most relevant document is chosen to be the representant. A query-biased summary is generated from the body of each of the group repesentative document. All the summary generation process occurs in the news summarizer component.
7. Finally all the results generated by the application model are exposed in the view layer in the user side.

Technologies. The system is developed as an Java EE 5 web application. For the crawling task we chose Nutch, the indexing in our system is also based on another Apache Software Foundation product, Lucene.

3 Research Issues

NowOnWeb shows innovative solutions to the three main problems of an IR news system, i.e., news recognition and extraction, redundancy detection and summary generation. Next we will introduce these issues and the designed solutions.

3.1 News Recognition and Extraction

In this case the problem is the existence in the set of crawled pages of a lot of not-article documents. Also into the article pages there are parts of them that are not-desired content, like advertisements, menus, galleries, etc.

This problem was previously addressed adapting web data extraction techniques. Results are not satisfactory due to the low quality of the extraction process outcomes, the necessity of human intervention or the high computational complexity required by these methods.

The extraction process could be considered as a two steps mechanism: news recognition and news extraction. The first one deals with the identification of news articles pages within a collection of heterogeneous web documents where there is not desired content (e.g. section pages, headlines, advertisements...). The second phase consists in, given a document identified like a *news page*, to extract the fields of the article, i.e., the title, the body and the image.

The main problem of existing methods is their complexity [7]. The high computational cost arises from the adaptation of general data extraction techniques [3] without taking advantage of the properties of the specific domain.

To overcome these difficulties we propose a method for new recognition and extraction from a set of web documents, based on domain specific heuristics, that takes advantage of the common characteristics of the articles structure, resulting in a efficient and effective algorithm that solves the news data extraction

problem. Both steps, news recognition and extraction, are considered together by our heuristics. In Figure 2 we show examples and counterexamples of news pages. News data extraction addresses the problem of obtaining the different

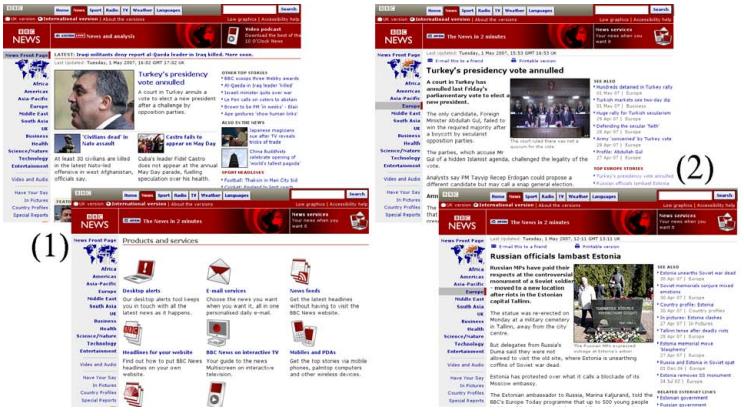


Fig. 2. Not-news pages (1) and news pages (2) present in the news web sites

parts of a news article, that are the title, the news body, and the image.

We propose a set of heuristics to identify and extract the article, or reject a not-news webpage:

1. *News are composed of paragraphs that are next one each other.*
2. *Paragraphs have a minimum size. News also have a minimum size.*
3. *Paragraphs are mostly text. Only styling markup and hyperlinks are allowed in paragraphs. Markup abounds in not desired content.*
4. *A low number of hyperlinks are allowed in paragraphs. A high number indicates not desired content, probably section pages.*

We also developed a set of heuristics to identify the title of the articles and the image, if it exists. Basically the title is on the top of the news body and has a highlighted typing style, usually it also substantially matches with the content of the HTML *title* tag. Among all the images present in a webpage we have to choose which one is relevant with the article; for this task we consider the position next or inside the news body, the content of the HTML *alt* tag, the size of the picture, the image format and the *url*.

3.2 Redundancy Detection

When NowOnWeb retrieves the articles that match the user query, it is not suitable to overload him with dozens of repeated articles about the same event from the different sources. As solution we adapted the tecnique for redundancy detection in Information Filtering proposed in [10].

Our method takes as input a ranking of articles in base of its relevance to the query. This list is the result of the search proccess and the score assigned to each document is the output of a classic similarity meassurement showed next:

Given a query q and a document d the *score* of the document d respect to the query q is defined by:

$$score(q,d) = \sum_{t_{in}q}(tf(t,d) \times idf(t)^2 \times boost(t,q) \times boost(t_{field},d)$$

$$\times norm_length(t_{field},d)) \times coord(q,d) \times norm_query(square_summatory)$$

Where:

$$square_summatory = \sum_{t_{in}q}(idf(t) * boost(t,q))^2$$

and t is a term of q, $tf(t,d)$ is the frequency of t in d, $idf(t)$ is the inverse frequency of the term calculated over the whole index, $boost(t_{field},d)$ is a scoring factor that depending of the document field where the term is found (url, *anchor*, title...) assigns more or less importance to the term. The score alo includes useful normalization factors to avoid the extra weight of the long documents and queries.

The redundancy of two documents d_t and d_i is based on the cosine distance:

$$R(d_t \mid d_i) = \frac{\sum_{k=1}^{m} t_k(d_t) \times t_k(d_i}{\sqrt{\sum_{k=1}^{m} t_k(d_t)^2 \sum_{k=1}^{m} t_k(d_i)^2}}$$

Where m is the number of terms in the lexicon and t_k is the result of the weighting function for the term t_k of the document d:

$$t_k(d) = tf(t_k,d) \times idf(t_k) * norm_{length}(d)$$

Using the cosine distance as the redundancy measurement, our algorithm groups similar news as follows. The first document in the ranking d_1 is selected as the representant of the first redundancy group. For the next documents in the ranking their redundancy with respect to d_1 is computed. If this value is over a redundancy threshold, this document is included in the same redundancy group, otherwise this document is selected as the representant of a new redundancy group. For the following document in the rank we have to repeat the proccess but computing the redundancy over the representants of all the group already existing, in the creation order and so on.

3.3 Summary Generation

Our system deals with summary generation with a technique based on the extraction of relevant sentences. Query-biased summaries are generated in retrieval time. The steps to get a query-biased summary are the following. Firstly we

extract from the article body the array of sentences. For each sentence we score it with the relevance formula [1] specified as:

Given a sentece s and a query q the relevance of s respec t to q is:

$$R(s \mid q) = \sum_{t \in q} \log(tf(t,q)+1)\log(tf(t,s)+1)\log(\frac{n+1}{0.5+sf(t)})$$

Where:

$tf(t,s)$ is the frequency of term t in the sentence s.
$tf(t,q)$ is the frequency of term t in the query q.
$sf(t)$ is the total number of sentence where t is present.
n is the number of document sentences.

Top scored sentences are selected until to fill the summary desired length, in our case the 30% of the article original size [5]. In the summary the sentences are reordered to mantain the original order in the article.

4 Related Work

News search is one of the most succesful activity in Internet, because of that the main internet companies offer this service. Many commercial news services like Google News, Yahoo News or MSN Newsbot maybe allow to the user searching in a huge repository of news due to their capacity, but, at the time of writing this paper, they do not produce summaries.

These sites show to the user clusters of the news represented by their title. We think that summaries are more convenient to facilitate the information access to the users avoiding to them to have to read the whole articles from the original sources.

In the academic world, on the other hand, we only can find a few of works. *NewsInEssence* [8] is a system for news clustering search and summarization of the Michigan University. *NewsBlaster* [6] is a development of Columbia University that also provides news clustering summarization, i.e., multidocumental summarization of the articles about specific events. But both of them are centered in event summarization not in articles summarization, so in that sense our approach offers a more natural way to access to the information. In these sense, *NowOnWeb* has a reduced query response time, faster contents update and also allows dinamically source adding.

5 Conclusions and Future Work

In this paper we presented *NowOnWeb* that had become in a full news search and summarization web-application. In the development we confront the usual problems of a NewsIR system for which we proposed a set of solutions expressed as algorithms that resulted suitable. Mainly we carry out a new news recognition and extraction algorithm that being generic, in the sense of source independent, allowed

us the dinamic adding of new news publishers; a redundancy detection strategy based on clasical redundancy measurements like the cosine distance, and a summary generation technique based in extraction of relevance sentences that results in a good approach to produce coherent news summaries of high quality.

We maintain *NowOnWeb* in continuous development, so as further work the main points to research are: architectural system improvements to reduce the query processing time, query logging storage in an efficient structure for query mining in retrieval time and for further offline processing, and evaluation of the news recognition and extraction module.

Acknowledgements. The work reported here was cofunded by the "Secretaría de Estado de Universidades e Investigación" and FEDER funds under the project MEC TIN2005-08521-C02-02 and "Xunta de Galicia" under project PGIDIT06PXIC10501PN. The authors also want to acknowledge the additional support of the "Galician Network of NLP&IR" (2006/23).

References

1. Allan, J., Wade, C., Bolivar, A.: Retrieval and novelty detection at the sentence level. In: SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 314–321. ACM Press, USA (2003)
2. Cafarella, M., Cutting, D.: Building nutch: Open source search. Queue 2(2), 54–61 (2004)
3. Crescenzi, V., Mecca, G.: Automatic information extraction from large websites. J. ACM 51(5), 731–779 (2004)
4. Hatcher, E., Gospodnetic, O.: Lucene in Action (In Action series). Manning Publications Co., Greenwich, CT, USA (2004)
5. Hovy, E.: Text Summarization. In: Mitkov, R. (ed.) The Oxford Handbook of Computational Linguistics, ch.32, pp. 583–598 (2005)
6. McKeown, K.R., Barzilay, R., Evans, D., Hatzivassiloglou, V., Klavans, J.L., Nenkova, A., Sable, C., Schiffman, B., Sigelman, S.: Tracking and summarizing news on a daily basis with Columbia's Newsblaster. In: Proceedings of the Human Language Technology Conference (2002)
7. Reis, D.C., Golgher, P.B., Silva, A.S., Laender, A.F.: Automatic web news extraction using tree edit distance. In: WWW '04: Proceedings of the 13th international conference on World Wide Web, pp. 502–511. ACM Press, New York (2004)
8. Radev, D., Otterbacher, J., Winkel, A., Blair-Goldensohn, S.: NewsInEssence: Summarizing online news topics. Commun. ACM 48(10), 95–98 (2005)
9. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Commun. ACM 18(11), 613–620 (1975)
10. Zhang, Y., Callan, J., Minka, T.: Novelty and redundancy detection in adaptive filtering. In: SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 81–88. ACM Press, USA (2002)

Verification of Program Properties Using Different Theorem Provers: A Case Study*

J. Santiago Jorge, Victor M. Gulias, and Laura M. Castro

MADS group, Department of Computer Science, Universidade da Coruña
Campus de Elviña, s/n, 15071 A Coruña Spain
{sjorge,gulias,lcastro}@udc.es
www.madsgroup.org

Abstract. This paper explores the use of theorem provers to certify particular properties of software. Two different proof assistants are used to illustrate the method: Coq and Pvs. By comparing two theorem provers, conclusions about their suitability can be stated. The selected scenario is part of a real-world application: a distributed Video-on-Demand server. The development consists on two steps: first, the definition of a model of the algorithm to be studied in the proof assistants; second, the development and proving of the theorems.

1 Introduction

Showing that a program is correct is an important activity in the development of computer programs. *Formal methods* [1,2] employ mathematical proof procedures to verify a system with respect to its specifications. They complement traditional debugging techniques assuring that some relevant property holds in the program. There are two major approaches to formal methods: *model checking*, which proves that every possible state of the program satisfies a specification, and *theorem proving*, which derives a proof establishing that the program satisfies a specification.

In this work, we explore the use of theorem provers to certify relevant properties of software [3,4]. Theorem provers are usually based on a logic, like first-order logic, higher-order logic or a version of set theory, that provides a framework for the formalisation of mathematics. Interactive proof assistants require a human user to give hints to the system. Depending on the degree of automation, significant proof tasks can be performed automatically. In our study, Coq [5] and Pvs [6] are going to be used as theorem provers. However, the proofs presented in this work can also be easily constructed with similar tools. By comparing two theorem provers, conclusions about their suitability can be stated.

Our case study is part of a real-world application: a distributed *Video-on-Demand* server [7], implemented with the concurrent functional programming language Erlang [8]. Functional languages [9,10] allow us to develop programs which will submit to logical analysis quite easily.

* Supported by MEC TIN2005-08986 and Xunta de Galicia PGIDIT05TIC10503PR.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 233–240, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The development consists on two steps: first, the definition, in the Coq and Pvs proof assistants, of the studied algorithm and some relevant properties to be proved; second, the coding of the theorems. The disadvantage of this approach is that it does not guarantee the correctness of the actual code, but some abstract *model* of it; however this helps to increase the reliability on the system [2].

The paper is organised as follows: First, an overview of the distributed Video-on-Demand server VoDKA is shown and the portion of software to be verified is put in context. Then, section 3 presents the Coq and Pvs models used to describe the algorithm. After that, section 4 certifies the correctness of the program with both theorem provers. Finally, we expose our conclusions.

2 Case Study: The Video-on-Demand Server VoDKA

A *Video-on-Demand* server is a system that provides video services to several clients simultaneously. This servers are complex distributed systems that must satisfy some critical requirements, including large storage capacity, many concurrent users, high bandwidth, reliability, scalability, adaptability to the underlying topology, and low cost.

2.1 The VoDKA Project

VoDKA (*Video-on-Demand Kernel Architecture*, `http://madsgroup.org/vodka`) is a project devoted to provide broadband-quality video-on-demand services to cable clients. Clusters built from cheap off-the-shelf components are proposed as an affordable solution for the huge demand of resources of the system. Cluster resources were organised as a hierarchical storage system, identifying three different layers: (a) *Repository level*, to store all the available media; (b) *Cache level*, in charge of storing videos read from the repository level, before being streamed; and (c) *Streaming level*, a front-end in charge of protocol adaption and media streaming to the final client. The concurrent functional programming language Erlang [8] was selected for implementing this design.

2.2 The Cache Subsystem of the Video-on-Demand Server

The piece of software to be verified in this work is part of VoDKA's cache subsystem: the block allocation algorithm. The cache layer provides a large aggregate throughput that alleviates usual deficiencies (in terms of bandwidth and concurrency) of the repository level. It is composed by a set of physical nodes each running agents and modules implementing different algorithms and policies (some quite tricky and error prone).

The local disk of each node of the cache level acts as a *local cache*, divided into fixed-size blocks which store fragments of a media object. In order to attend a request, a media object is firstly looked up in local caches, avoiding the access to the repository layer if possible. If a media object must be fetched to cache (because of a cache miss), enough space must be booked to load the object from the storage. In order to release some blocks, we have to assure that these blocks

are not in use by other pending tasks. As locality in cache is important for performance, blocks of the same media object are in close proximity and we can speak of *block intervals*.

3 Algorithm Modelling

3.1 Block Interval Representation

A block interval is represented as a triple (a, b, x): the interval between blocks a and b (inclusive) has x pending tasks. A list is used to store the whole sequence of block intervals: $[(a_1, b_1, x_1), (a_2, b_2, x_2), \ldots, (a_n, b_n, x_n)]$. Efficiency is achieved if the intervals are kept sorted $\forall i, a_i \leq b_i \wedge b_i < a_{i+1}$. Space is saved if the sequence is kept compact, i.e. $\forall i, x_i \neq 0$, and $(x_i \neq x_{i+1}) \vee ((x_i = x_{i+1}) \wedge (b_i + 1 < a_{i+1}))$

Several definitions are needed before translating algorithms on blocks, intervals and sequences into a model. We show both Coq and Pvs approaches to highlight the differences between these provers. For an introduction to Coq, refer to [11,12]. For a tutorial relating to Pvs, refer to [13].

In Coq a *block interval* is defined as a triple of natural numbers including the initial and final block and the number of pending requests requiring the interval.

```
Inductive interval: Set:=  tuple: nat -> nat -> nat -> interval.
```

`Interval` is an inductive definition with one constructor, `tuple`, that gets three natural numbers to create an interval. A *sequence* is then represented as a list of intervals using an inductive type. The constructors are `empty` and `make`. The former represents an empty sequence, the latter gets an interval and a sequence and it returns a new sequence.

```
Inductive seq: Set:= empty: seq | make: interval->seq->seq.
```

In Pvs, the definitions are analogous to the ones used in the Coq model: an interval is a tuple of three elements, and a sequence is a list of intervals.

```
interval: TYPE = [nat, nat, nat]
seq: TYPE = list[interval]
```

3.2 Modelling the Behaviour of Block Allocator

The core of the block allocation algorithm is the `add` function which sums up new block requests. Testing showed up several bugs but, unfortunately, this does not guarantee software correctness. In order to certify the correctness of the program, we write a model from the actual Erlang code in the proof assistants.

Figure 1(left) shows (part of) the Coq model. The function is defined using structural recursion over the second argument: the interval sequence. This is denoted by `{struct l}`. In order to be accepted, this argument has to decrease structurally in the recursive calls. Figure 1(right) shows the same translation into Pvs. There is a `MEASURE` function specified in order to guarantee that the definition is total. Pvs uses the `MEASURE` to generate an obligation to ensure that

```
Fixpoint add (n:nat)(l:seq){struct l}:seq:=
 match l with
 |empty => make (tuple n n (S O)) empty
 |make (tuple a b x) l' =>
   match le_lt_dec a n with
   |left _ =>
     match le_lt_dec n b with
     |left _ =>
       match eq_nat_dec a n with
       |left _ =>
         match eq_nat_dec n b with
         |left _ =>make (tuple a b (S x)) l'
         |right _ =>
           make (tuple a a (S x))
             (make (tuple (S a) b x) l')
         end
       |right _ =>
         match eq_nat_dec n b with
         |left _ =>
           make (tuple a (pred b) x)
             (make (tuple b b (S x)) l')
         |right _ =>
           make (tuple a (pred n) x)
             (make (tuple n n (S x))
               (make (tuple (S n) b x) l'))
         end
       end
     |right _ =>make (tuple a b x) (add n l')
     end
   |right _ =>make (tuple n n (S O)) l
   end
 end.
```

```
add(n:nat,l:seq): RECURSIVE seq =
 CASES l OF
  null: make(tuple(n,n,1),null),
  cons(i,l1):
   IF i`1 <= n
   THEN
     IF n <= i`2
     THEN
       IF i`1 = n
       THEN
         IF n = i`2
         THEN make(tuple(i`1,i`2,i`3+1),l1)
         ELSE make(tuple(i`1,i`1,i`3+1),
                make(tuple(i`1+1,i`2,i`3),l1))
         ENDIF
       ELSE
         IF n = i`2
         THEN make(tuple(i`1,i`2-1,i`3),
                make(tuple(i`2,i`2,i`3+1),l1))
         ELSE make(tuple(i`1,n-1,i`3),
                make(tuple(n,n,i`3+1),
                 make(tuple(n+1,i`2,i`3),l1)))
         ENDIF
       ENDIF
     ELSE make(i,add(n,l1))
     ENDIF
   ELSE make(tuple(n,n,1),l)
   ENDIF
  ENDCASES
 MEASURE length(l)
```

Fig. 1. Definition of `add` in Coq (left) and Pvs (right)

the function is total. This proof obligation is generated as a *Type-Correctness Condition* (TCC).

As shown, Coq and Pvs have different approaches to the definition of primitive recursive functions. Coq is more restrictive, in the sense that it demands the use structural induction on one argument of the function. Pvs's solution is more flexible.

4 Proof Development with the Explored Provers

4.1 Proving Properties of `add` with Coq

First of all, an auxiliary function `nth` is defined to return the number of pending requests on the nth block. This function, certified in [3], is going to be used to state the specification.

The `add` function must satisfy that, in the resulting interval sequence, the number of requests over the requested block is incremented by one, while all the other blocks remain unchanged.

$$\texttt{nth n (add i l)} = \begin{cases} \texttt{nth n l} & \text{if } \texttt{i} \neq \texttt{n} \\ 1 + \texttt{nth n l} & \text{if } \texttt{i} = \texttt{n} \end{cases}$$

This property is splitted into two theorems provided that `i` equals `n` or not.

```
Theorem nth_add_1: forall (l: seq) (n m: nat),
  n = m -> nth n (add m l) = S (nth n l).
```

The proof of a property is carried out applying successive tactics which solve the goal or transform it into new subgoals. The proof process ends when no subgoals remain unproved.

Both the definition of `add` and `nth` are carried out by structural recursion on the sequence of block intervals. Thus, we proceed by induction on `l`. In the proof, case analyses, β-reductions and equational reasonings are performed following the structure of function `add`.

The second theorem to be proved is quite similar (though larger):

```
Theorem nth_add_2 : forall (l: seq) (n m: nat),
  n <> m -> nth n (add m l) = nth n l.
```

4.2 Proving Properties of `add` in PVS

As done in Coq, we define an auxiliary function `nth` to state the specification. The resulting theorems are formulated as:

```
nth_add_1: THEOREM FORALL (m, n: nat), (l: seq):
  m = n => nth(n, add(m, l)) = 1 + nth(n, l)
nth_add_2: THEOREM FORALL (m, n: nat), (l: seq):
  m /= n => nth(n, add(m, l)) = nth(n, l)
```

In this case, Pvs proofs are carried out by the system *automagically* using the advanced tactic `induct-and-simplify`. A great deal of work is carried out by Pvs if compared with Coq version.

4.3 Assuring the Canonical Form of Interval Sequences

The representation of interval sequences should be canonical (in the sense that two different representations always correspond to two different objects) to get better space and time behaviour. We should demonstrate that the application of `add` to an interval sequence in canonical form always delivers a new canonical sequence.

First, we present the property that states that intervals in a sequence are in ascendent order. The predicate `ascend` asserts whether a sequence of intervals $[(a_1, b_1, x_1), \ldots, (a_n, b_n, x_n)]$ is sorted in ascending order, that is, it shows whether $\forall i, a_i \leq b_i \wedge b_i < a_{i+1}$ is true or not.

In what remains, Coq code is presented in the first column and Pvs code in the second one.

```
Inductive ascend : seq->Prop :=
  ascend1 : ascend empty
| ascend2 : forall a b x:nat,
   le a b ->
     ascend (make (tuple a b x) empty)
| ascend3 : forall (l:seq)(a b x c d y:nat),
   le a b -> lt b c ->
     ascend (make (tuple c d y) l) ->
       ascend (make (tuple a b x)
                 (make (tuple c d y) l)).
```

```
ascend(l: seq): RECURSIVE bool =
  CASES l
    OF null: TRUE,
       cons(x, xs):
         CASES xs
           OF null: x`1 <= x`2,
              cons(y, ys): ascend(xs) AND
                     x`1 <= x`2 AND x`2 < y`1
           ENDCASES
    ENDCASES
   MEASURE length(l)
```

This predicate is not enough to define the canonical representation because the same sequence could be represented in different ways, e.g. the sequence `[(2,2,1),(3,5,1),(8,9,2)]` could also be represented by `[(2,5,1),(8,9,2)]` which is more compact. The representation must also be *packed*, i.e. it must hold that $\forall i, a_i \leq b_i \wedge ((x_i \neq x_{i+1} \wedge b_i < a_{i+1}) \vee (x_i = x_{i+1} \wedge b_i + 1 < a_{i+1}))$.

```
Inductive packed : seq->Prop :=
  packed1 : packed empty
| packed2 : forall a b x:nat, le a b ->
    packed (make (tuple a b x) empty)
| packed3 : forall (l:seq) (a b x c d y:nat),
    packed (make (tuple c d y) l) ->
      le a b -> x<>y -> lt b c ->
        packed (make (tuple a b x)
                  (make (tuple c d y) l))
| packed4 : forall (l:seq) (a b x c d y:nat),
    packed (make (tuple c d y) l) ->
      le a b -> lt (S b) c ->
        packed (make (tuple a b x)
                  (make (tuple c d y) l)).
```

```
packed(l: seq): RECURSIVE bool =
  CASES l
    OF null: TRUE,
       cons(x, xs):
        CASES xs
          OF null: x‘1 <= x‘2,
             cons(y, ys):
               packed(xs) AND x‘1 <= x‘2 AND
                 ((x‘3 /= y‘3 AND x‘2 < y‘1)
                   OR (x‘2 + 1) < y‘1)
        ENDCASES
   ENDCASES
   MEASURE length(l)
```

The output of `add` holds the predicate `ascend` but it does not hold `packed`. In order to satisfy the specification, a new function `pack` is defined to convert the result of `add`.

```
Fixpoint pack_aux (i:interval) (l:seq)
  {struct l} : seq:= match i with
  tuple a b x=> match l with
     empty=> make i empty
    |make (tuple c d y) l' =>
      match eq_nat_dec x y with
       left _=> match eq_nat_dec (S b) c with
          left _=> pack_aux (tuple a d x) l'
         |right _=>
           make i (pack_aux (tuple c d y) l')
        end
      |right _=>
        make i (pack_aux (tuple c d y) l')
      end
    end
  end.
Definition pack (l:seq):seq := match l with
    | empty => empty
    | make i l' => pack_aux i l'
  end.
```

```
packAux(i: interval, l: seq): RECURSIVE seq =
  CASES l
    OF null: make(i,empty),
       cons(x, xs):
         IF i‘3 = x‘3 THEN
           IF i‘2 + 1 = x‘1 THEN
             packAux(tuple(i‘1,x‘2,i‘3),xs)
           ELSE make(i,packAux(x,xs))
           ENDIF
         ELSE make(i,packAux(x,xs))
         ENDIF
    ENDCASES
   MEASURE length(l)

pack(l: seq): seq =
    CASES l OF null: empty,
               cons(x, xs): packAux(x, xs)
      ENDCASES
```

To complete the canonical representation, we need a predicate which guarantees that every block in the sequence has at least one request.

```
Inductive strict_pos: seq->Prop:=
  strict_pos1: (strict_pos empty)
| strict_pos2: forall (l:seq) (a b x: nat),
    strict_pos l -> lt 0 x ->
      strict_pos (make (tuple a b x) l).

Inductive canonical: seq->Prop:=
  canonical1: forall (l:seq), (packed l) ->
    (strict_pos l) -> (canonical l).
```

```
strictPos (l: seq): RECURSIVE bool =
  CASES l OF null: TRUE,
     cons(x, xs): strictPos(xs) AND x‘3 > 0
    ENDCASES
   MEASURE length(l)

canonical(l: seq): bool =
   packed(l) AND strictPos(l)
```

We should demonstrate that `pack` returns an interval sequence equivalent to the one it gets as argument as long as it is sorted in ascendent order. It should

be mentioned that in PVS the powerful `induct-and-simplify` command fails in the next lemmas, so we have to relay on simpler tactics.

```
Lemma nth_pack : forall (l:seq) (n:nat),
 ascend l -> nth n l = nth n (pack l).
```

```
nth_pack: LEMMA FORALL (l: seq)(n: nat):
  ascend(l) => nth(n, l) = nth(n, pack(l))
```

Now, it is proved that the result of `add` is a sequence sorted in ascendent order and that the result of the application of `pack` on an ascendent sequence delivers a packed one.

```
Lemma ascend_add : forall (l:seq) (n:nat),
  ascend l -> ascend (add n l).
Lemma packed_pack: forall (l:seq),
  (ascend l) -> (packed (pack l)).
```

```
ascend_add: LEMMA FORALL (l: seq)(n: nat):
  ascend(l) => ascend(add(n, l))
packed_pack: LEMMA FORALL (l: seq):
  ascend(l) => packed(pack(l))
```

It is also proved that `add` and `pack` do not leave any block without requests.

```
Lemma strict_pos_add : forall (l:seq)(n:nat),
  strict_pos l -> strict_pos (add n l).
Lemma strict_pos_pack : forall (l:seq),
  strict_pos l -> strict_pos (pack l).
```

```
strictPos_add: LEMMA FORALL (l: seq)(n: nat):
  strictPos(l) => strictPos(add(n, l))
strictPos_pack: LEMMA FORALL (l: seq):
  strictPos(l) => strictPos(pack(l))
```

Finally, we should state and prove the law which sets that given a sequence in canonical form, adding a request to any block with `add`, and applying `pack` after that, gets a new sequence in canonical form.

```
Theorem can_pack_add: forall (l:seq)(n:nat),
  canonical l -> canonical (pack (add n l)).
intros l n h.
inversion_clear h as [l' hpack hpos].
constructor.

2 subgoals
  l : seq
  n : nat
  hpack : packed l
  hpos : strict_pos l
  ============================
   packed (pack (add n l))
subgoal 2 is:
 strict_pos (pack (add n l))

apply packed_pack.
apply ascend_add.
apply packed_ascend. auto...
apply strict_pos_pack.
apply strict_pos_add. auto...
Qed.
```

```
can_pack_add: THEOREM FORALL (l:seq)(n:nat):
  canonical(l) => canonical(pack(add(n, l)))

(skosimp*)
(expand "canonical")
(flatten)
(split)

this yields  2 subgoals:
canonical_pack_add.1 :
[-1]  packed(l!1)
[-2]  strictPos(l!1)
  |-------
1   packed(pack(add(n!1, l!1)))

(use "packed_pack") (split -1)
(use "ascend_add") (split -1)
(use "packed_ascend") (split -1)
(use "strictPos_pack") (split -1)
(use "strictPos_add") (split -1)

Q.E.D.
```

5 Conclusions

The distributed *Video-on-Demand* server used for our study is a *real-world* application of a functional language. We have expounded separately the formal verification of part of the functionality of the server. The verification is not done on the real system but on some abstract *model* of it in COQ and PVS. It does not guarantee the correctness of the actual code, but helps to increase the reliability on the system. Increasing software reliability is the final goal of formal methods.

In general, the elaboration of proofs is based on the syntactic structure of the program. Complexity is managed by dividing the problem into smaller problems using auxiliary laws. Coq and Pvs have different philosophies: Coq formalises higher-order logic with inductive types, which gives a rigorous notion of proof and the possibility of extracting functional programs from the algorithmic content of the proof. Pvs does not have such a rigorous notion of proof but it incorporates powerful automatic theorem-proving capabilities that reduce the amount of work for the developer. In this sense, Pvs seems to be more useful for software engineers and easier to be integrated into software development life cycle.

Large systems are difficult to study as a whole and therefore, they must be decomposed into small pieces in order to study each one separately; later, when we combine other partial results, conclusions on the whole system could be drawn. In this sense, the future of program verification heads for the general recommendation of obtaining certified software libraries.

References

1. Clarke, E.M., Wing, J.M.: Formal methods: state of the art and future directions. ACM Computing Surveys 28, 626–643 (1996)
2. Peled, D.A.: Software Reliability Methods. Springer, Heidelberg (2001)
3. Jorge, J.S.: Estudio de la verificación de propiedades de programas funcionales: de las pruebas manuales al uso de asistentes de pruebas. PhD thesis, University of A Coruña, Spain (2004)
4. Jorge, J.S., Gulías, V.M., Cabrero, D.: Certifying Properties of Programs using Theorem Provers. In: Verification, Validation and Testing in Software Engineering, pp. 220–267. Idea Group Publishing, USA (2007)
5. Bertot, Y., Casteran, P.: Interactive Theorem Proving and Program Development, Coq'Art: The Calculus of Inductive Constructions. Springer, Heidelberg (2004)
6. Owre, S., Shankar, N., Rushby, J.M.: The PVS Specification Language. Computer Science Laboratory, SRI International (1995)
7. Gulías, V.M., Barreiro, M., Freire, J.L.: VODKA: Developing a video-on-demand server using distributed functional programming. Journal of Functional Programming 15, 403–430 (2005)
8. Armstrong, J., Virding, R., Wikström, C., Williams, M.: oncurrent Programming in Erlang, 2nd edn. Prentice Hall, Prentice Hall (1996)
9. Bird, R., Wadler, P.: Introduction to Functional Programming. Prentice Hall, Prentice Hall (1988)
10. Hudak, P.: Conception, evolution, and application of functional programming languages. ACM Computing Surveys 21, 359–411 (1989)
11. Giménez, E.: A tutorial on recursive types in Coq. Technical report, INRIA, for Coq V6.2 (1998)
12. Huet, G., Kahn, G., Paulin-Mohring, C.: The Coq proof assistant: A tutorial, v8.0. Technical report, INRIA (2004)
13. Crow, J., Owre, S., Rushby, J., Shankar, N., Srivas, M.: A tutorial introduction to PVS. In: WIFT'95 Workshop on Industrial-Strength Formal Specification Techniques (1995)

Temporal Equilibrium Logic: A First Approach*

Pedro Cabalar and Gilberto Pérez Vega

Dept. Computación, Corunna University Spain
{cabalar,gperez}@udc.es

Abstract. In this paper we introduce an extension of Equilibrium Logic (a logical characterisation of the Answer Set Semantics for logic programs) consisting in the inclusion of modal temporal operators, as those used in Linear Temporal Logic. As a result, we obtain a very expressive formalism that allows nonmonotonic reasoning for temporal domains. To show an example of its utility, we present a translation of a language for reasoning about actions into this formalism.

1 Introduction

One of the areas of Logic Programming that has attracted more research interest during the last decade is *Answer Set Programming* (ASP). The origins of this area come from the achievement of a suitable semantics (first called *stable models* [1] and later generalized into *answer sets* [2]) for interpreting default negation in *normal* logic programs[1]. The availability of this semantics allowed considering logic programs as a general knowledge representation formalism, especially for solving default and non-monotonic reasoning problems. Under this new focusing, a process of gradual syntactic extensions followed, most of them eventually captured by [3], a completely logical characterisation of answer sets into the non-monotonic formalism of *Equilibrium Logic*. This approach offers a standard logical interpretation for any arbitrary theory, removing any syntactic restriction while generalising all previous extensions of answer set semantics.

Apart from with this nice logical characterisation, the recent success of ASP is mostly due to several existing tools (see [4]) offering a high efficiency degree for computing answer sets of a normal logic program (an NP-complete problem). The price to be paid for this efficiency is the impossibility of handling function symbols, restriction that guarantees a finite domain and allows grounding the program into a finite set of propositional rules. *Planning* for transition systems constitutes one of the most popular application areas of ASP. Typically, the logic program is used to encode the initial state, the transition rules and the goal description. Default negation plays here a crucial role, as it allows avoiding the frame problem [5] by representing the default rule of *inertia*: "if a fluent (or

* This research is partially supported by Spanish Ministry MEC, research project TIN-15455-C03-02.

[1] Those with rules containing an atom in the head and a body consisting of a conjunction of literals.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 241–248, 2007.
© Springer-Verlag Berlin Heidelberg 2007

temporal variable) is not known to have changed, it will remain unchanged by default." The general planning problem lies in a harder class (PSPACE-complete) than the one covered by ASP solvers. This means in practice that, in order to solve planning problems, it is first required to *fix the length* l of the plan to be obtained. Usually, the plan search consists in gradually incrementing this length parameter $l = 0, 1, 2, 3, \ldots$ until a plan is obtained. A first obvious drawback of this approach is that it is not possible to establish when a given planning problem has *no solution* of any length at all. A second and more elaborated problem is that it is impossible to establish when two descriptions of the same transition system are *strongly equivalent*, i.e., when they will behave in the same way regardless any additional rules we include and any path length we consider.

In the paper we consider an extension of Equilibrium Logic (and so, of logic programming) to deal with modal operators as those handled in Linear Temporal Logic (LTL) [6] [7]: $\Box$ (forever), $\Diamond$ (eventually), $\bigcirc$ (next), $\mathcal{U}$ (until) and $\mathcal{W}$ (waiting for). Equilibrium Logic is defined as a model minimisation on top of a monotonic framework, the non-classical intermediate logic of *here-and-there* (HT) [8]. The temporal extension will be defined in a similar way, introducing first the temporal extension of HT (called THT), and then a minimisation criterion. For space reasons, we omit the description of Equilibrium logic, as it can be defined as a particular case of its temporal extension.

The paper is organised as follows. Section 2 describes the semantics of THT and Section 3 defines a models minimisation criterion for THT models that leads to Temporal Equilibrium Logic (TEL). In the next section, we show the utility of TEL providing the translation of a language for reasoning about actions. Finally, Section 5 contains the conclusions and future work.

2 Linear Temporal Here-and-There (THT)

The logic of *Linear Temporal Here-and-There* (THT) is defined as follows. We start from a finite set of atoms Σ called the *propositional signature*. A (temporal) *formula* is defined as any combination of the classical connectives $\wedge, \vee, \rightarrow, \bot$ with the (unary) temporal operators $\Box, \Diamond, \bigcirc$, the (binary) temporal operators $\mathcal{U}, \mathcal{W}$ and the atoms in Σ. A formula is said to be *non-modal* if it does not contain temporal operators. Negation is defined as $\neg\varphi \stackrel{def}{=} \varphi \rightarrow \bot$ whereas $\top \stackrel{def}{=} \neg\bot$. As usual, $\varphi \leftrightarrow \psi$ stands for $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. We also allow the abbreviation $\bigcirc^i\varphi \stackrel{def}{=} \bigcirc(\bigcirc^{i-1}\varphi)$ for $i > 0$ and $\bigcirc^0\varphi \stackrel{def}{=} \varphi$.

A (temporal) *interpretation* M is an infinite sequence of pairs $\langle H_i, T_i\rangle$ with $i = 0, 1, 2, \ldots$ where $H_i \subseteq T_i$ are sets of atoms. For simplicity, given a temporal interpretation, we write $\overline{H}$ (resp. $\overline{T}$) to denote the sequence of pair components $H_0, H_1, \ldots$ (resp. $T_0, T_1, \ldots$). Using this notation, we will sometimes abbreviate the interpretation as $M = \langle\overline{H}, \overline{T}\rangle$. An interpretation $M = \langle\overline{H}, \overline{T}\rangle$ is said to be total when $\overline{H} = \overline{T}$. We say that an interpretation $M = \langle\overline{H}, \overline{T}\rangle$ satisfies a formula φ at some sequence index i, written $M, i \models \varphi$, when any of the following hold:

1. $M, i \models p$ if $p \in H_i$, for any atom p.
2. $M, i \models \varphi \wedge \psi$ if $M, i \models \varphi$ and $M, i \models \psi$.
3. $M, i \models \varphi \vee \psi$ if $M, i \models \varphi$ or $M, i \models \psi$.
4. $\langle \overline{H}, \overline{T} \rangle, i \models \varphi \rightarrow \psi$ if $\langle X, \overline{T} \rangle, i \not\models \varphi$ or $\langle X, \overline{T} \rangle \models \psi$ for all $X \in \{\overline{H}, \overline{T}\}$.
5. $M, i \models \bigcirc \varphi$ if $M, i+1 \models \varphi$.
6. $M, i \models \Box \varphi$ if for all $j \geq i$, $M, j \models \varphi$.
7. $M, i \models \Diamond \varphi$ if there exists some $j \geq i$, $M, j \models \varphi$.
8. $\varphi \, \mathcal{U} \, \psi$ if $\exists j \geq i, \quad M, j \models \psi$ and $\forall k$ s.t. $i \leq k < j, \quad M, k \models \varphi$
9. $\varphi \, \mathcal{W} \, \psi$ if either $M, i \models \varphi \, \mathcal{U} \, \psi$ or, for all $j \geq i, \;\; M, j \models \varphi$.

We assume that a finite sequence $\langle H_0, T_0 \rangle \ldots \langle H_n, T_n \rangle$ with $n \geq 0$ is an abbreviation of the infinite sequence $\langle \overline{H}', \overline{T}' \rangle$ with $H'_i = H_i$, $T'_i = T_i$ for $i = 0, \ldots, n$ and $H'_i = H_n$, $T'_i = T_n$ for $i > n$.

The logic of THT is an orthogonal combination of the logic of HT and the (standard) linear temporal logic (LTL) [7]. When we restrict temporal interpretations to finite sequences of length 1, that is $\langle H_0, T_0 \rangle$ and disregard temporal operators, we obtain the logic of HT. On the other hand, if we restrict the semantics to total interpretations, $\langle \overline{T}, \overline{T} \rangle, i \models \varphi$ corresponds to satisfaction of formulas $\overline{T}, i \models \varphi$ in LTL.

A *theory* is any set of formulas. An interpretation M is a *model* of a theory Γ, written $M \models \Gamma$, if $M, 0 \models \alpha$, for all formula $\alpha \in \Gamma$. A formula φ is *valid* if $M, 0 \models \varphi$ for any M. The following are valid formulas in THT (and in LTL too):

$$\neg \Box \varphi \leftrightarrow \Diamond \neg \varphi \tag{1}$$
$$\neg \Diamond \varphi \leftrightarrow \Box \neg \varphi \tag{2}$$
$$\bigcirc \neg \varphi \leftrightarrow \neg \bigcirc \varphi \tag{3}$$
$$\bigcirc (\varphi \wedge \psi) \leftrightarrow \bigcirc \varphi \wedge \bigcirc \psi \tag{4}$$
$$\bigcirc (\varphi \vee \psi) \leftrightarrow \bigcirc \varphi \vee \bigcirc \psi \tag{5}$$
$$\bigcirc (\varphi \rightarrow \psi) \leftrightarrow (\bigcirc \varphi \rightarrow \bigcirc \psi) \tag{6}$$
$$\varphi \, \mathcal{U} \, \psi \leftrightarrow \varphi \, \mathcal{W} \, \psi \wedge \Diamond \psi \tag{7}$$
$$\varphi \, \mathcal{W} \, \psi \leftrightarrow \varphi \, \mathcal{U} \, \psi \vee \Box \varphi \tag{8}$$

Note that (3)-(6) allow shifting $\bigcirc$ through any combination of non-modal operators. In this way, any theory not containing $\Box, \Diamond, \mathcal{U}, \mathcal{W}$ can be transformed into a combination of non-modal connectives applied to expressions like $\bigcirc^i p$, where p is an atom. Notice also that, even though (1) and (2) are valid, we cannot further define $\Box$ in terms of $\Diamond$ or vice versa, as happens in LTL. For instance, as a countermodel of $\Box p \leftrightarrow \neg \Diamond \neg p$ in THT it suffices to take an interpretation $\langle \overline{H}, \overline{T} \rangle$ with all $T_i = \{p\}$ but some $H_i = \emptyset$.

We can alternatively represent any interpretation $M = \langle \overline{H}, \overline{T} \rangle$ as an infinite sequence of mappings V_i assigning a truth value to each atom $V_i : \Sigma \rightarrow \{0, 1, 2\}$ where 0 stands for false, 2 for true and 1 for undefined. We can define a valuation for any formula φ at any sequence index i, written $M_i(\varphi)$, by just considering which formulas are satisfied by $\langle \overline{H}, \overline{T} \rangle$ (which will be assigned 2), not satisfied by $\langle \overline{T}, \overline{T} \rangle$ (which will be assigned 0) or none of the two (which will be 1). Thus, the definitions in the previous section allow deriving the following conditions:

1. $M_i(p) \stackrel{def}{=} V_i(p)$
2. $M_i(\varphi \wedge \psi) \stackrel{def}{=} min(M_i(\varphi), M_i(\psi))$
3. $M_i(\varphi \vee \psi) \stackrel{def}{=} max(M_i(\varphi), M_i(\psi))$
4. $M_i(\varphi \rightarrow \psi) \stackrel{def}{=} \begin{cases} 2 & \text{if } M_i(\varphi) \leq M_i(\psi) \\ M(\psi) & \text{otherwise} \end{cases}$
5. $M_i(\bigcirc\varphi) \stackrel{def}{=} M_{i+1}(\varphi)$
6. $M_i(\Box\varphi) \stackrel{def}{=} min\{M_j(\varphi) \mid j \geq i\}$
7. $M_i(\Diamond\varphi) \stackrel{def}{=} max\{M_j(\varphi) \mid j \geq i\}$
8. $M_i(\varphi \,\mathcal{U}\, \psi) \stackrel{def}{=} \begin{cases} 2 \text{ if exists } j \geq i, M_j(\varphi) = 2 \\ \quad \text{and } M_k(\psi) = 2 \text{ for all } k \in [i, j-1] \\ 0 \text{ if for all } j \geq i : M_j(\varphi) = 0 \\ \quad \text{or } M_k(\psi) = 0 \text{ for some } k \in [i, j-1] \\ 1 \text{ otherwise} \end{cases}$
9. $M_i(\varphi \,\mathcal{W}\, \psi) \stackrel{def}{=} M_i(\varphi \,\mathcal{U}\, \psi \vee \Box\varphi)$

Using this alternative definition, an interpretation M satisfies a formula φ when $M_0(\varphi) = 2$.

3 Linear Temporal Equilibrium Logic (TEL)

We can now proceed to describe the models selection criterion that defines temporal equilibrium models. Given two interpretations $M = \langle \overline{H}, \overline{T} \rangle$ and $M' = \langle \overline{H}', \overline{T}' \rangle$ we say that M is *lower than* M', written $M \leq M'$, when $\overline{T} = \overline{T}'$ and for all $i \geq 0$, $H_i \subseteq H'_i$. As usual, $M < M'$ stands for $M \leq M'$ but $M \neq M'$.

Definition 1 (Temporal Equilibrium Model). *An interpretation M is a* temporal equilibrium model *of a theory Γ if M is a total model of Γ and there is no other $M' < M$, $M', 0 \models \Gamma$.* □

Note that any temporal equilibrium model is total, that is, it has the form $\langle \overline{T}, \overline{T} \rangle$ and so can be actually seen as an interpretation $\overline{T}$ in the standard LTL.

Example 1. The temporal equilibrium models of theory $\{\Diamond p\}$ correspond to LTL interpretations like $[\ \emptyset\ ]^*\ \{p\}\ [\ \emptyset\ ]^*$.

Proof. Take any a total interpretation $M = \langle \overline{T}, \overline{T} \rangle$ with some $T_i = \{p\}$. Clearly, $M, 0 \models \Diamond p$, but its minimality will depend on whether there exists another $j \neq i$ with $T_j = \{p\}$ in $\overline{T}$. Assume there exists such a j: then it is easy to see that the interpretation $\{\overline{H}, \overline{T}\}$ with $H_j = \emptyset$ and $\overline{H}_k = \overline{T}_k$ for $k \neq j$ is also a model of $\Diamond p$ (since $H_i = T_i = \{p\}$) whereas it is strictly lower, so it would not be a temporal equilibrium model. Assume, on the contrary, that T_i is the unique state containing p, so $T_k = \emptyset$ for all $k \neq i$. Then, the only possible smaller interpretation would be $\{\overline{H}, \overline{T}\}$ with $H_k = T_k = \emptyset$ for $k \neq i$ and $H_i = \emptyset$, but this is not a THT model of $\Diamond p$, as p would not occur in $\overline{H}$. □

Example 2. Consider the theory Γ just consisting of the formula $\Box(\neg p \rightarrow \bigcirc p)$. Its temporal equilibrium models correspond to LTL interpretations like $\emptyset \left[\{p\}\ \emptyset \right]^*$ or $\emptyset \left[\{p\}\ \emptyset \right]^* \{p\}$.

Proof. For an informal sketch, note that any solution with p true at T_0 will not be minimal. Once $\neg p$ is fixed at the initial state, we must get p at T_1 to satisfy the formula. Then p true at T_2 would not be minimal, so it must be false, and so on. □

Proposition 1. *Let Γ be a combination of non-modal connectives $\wedge, \vee, \neg, \rightarrow, \bot$ with expressions like $\bigcirc^i p$, being p an atom, and let n be the maximum value for i in all $\bigcirc^i p$ occurring in Γ. Then $\langle \overline{T}, \overline{T} \rangle$ is a temporal equilibrium model of Γ iff (1) $T_i = \emptyset$ for all $i > n$; and (2) $\langle X, X \rangle$ with $X = \bigcup_{i=0}^{n} \{\bigcirc^i p \mid p \in T_i\}$ is an equilibrium model of Γ, reading each '$\bigcirc^i p$' as a new atom in the signature.* □

That is, once $\Box, \Diamond, \mathcal{U}$ and $\mathcal{W}$ are removed, we can reduce TEL to (non-temporal) Equilibrium Logic for an extended signature with atoms like $\bigcirc^i p$.

4 Translation of an Action Description Language

In this section we briefly introduce an example of how TEL can be naturally used to translate action description languages. To this aim, we consider an encoding of language $\mathcal{B}$ [9], obtaining a similar translation to the one previously presented in [10] for a different temporal nonmonotonic formalism. Language $\mathcal{B}$ defines a two-sorted signature: the set $\mathcal{A}$ of action names and the set $\mathcal{F}$ of fluent names. By *fluent literal* we understand any fluent name F or its explicit negation, we denote[2] as $\overline{F}$. A *state* is a maximal consistent (that is, not containing both F and $\overline{F}$) set of fluent literals. An action description $D = Y \cup Z$ consists of a set Y of *dynamic laws* of the form (A **causes** L **if** φ) and a set Z of *static laws* of the form (L **if** φ) where A is an action, L a fluent literal and φ a conjunction of fluent literals. The semantics of $\mathcal{B}$ is defined as follows. A set s of literals is *closed under* Z iff for each (L **if** φ) $\in Z$ such that s satisfies φ, $L \in s$. Let $Cn_Z(s)$ be the smallest set of literals containing s that is closed under Z. For a given action description $D = Y \cup Z$, we define a transition system as a triple $\langle S, V, R \rangle$ where:

1. S is the set of all states that are closed under Z.
2. $V(P, s) = s(P)$ is a valuation function that assigns a truth value (*true* or *false*), to each fluent P at each state $s \in S$.
3. R is the transition relation containing all the triples $\langle s, A, s' \rangle$ such that $s' = Cn_Z(E(A, s) \cup (s \cap s'))$ being
   $E(A, s) \stackrel{def}{=} \{L : (A \textbf{ causes } L \textbf{ if } \varphi) \in Y$ and s satisfies φ

The translation into TEL is defined as follows. For each static or dynamic law $\phi \in D$ we define $\Pi(\phi)$ as:

$$\Pi(L \textbf{ if } \varphi) \stackrel{def}{=} \Box(\varphi \rightarrow L) \qquad \Pi(A \textbf{ causes } L \textbf{ if } \varphi) \stackrel{def}{=} \Box(A \wedge \varphi \rightarrow \bigcirc L)$$

[2] We use this notation to avoid confusion with default negation, represented as $\neg$ in this paper.

The TEL theory $\Pi(D)$ is defined as $\{\Pi(\phi) : \phi \in D\}$ plus the *inertia law*:

$$\Box(F \wedge \neg \bigcirc \overline{F} \rightarrow \bigcirc F) \tag{9}$$
$$\Box(\overline{F} \wedge \neg \bigcirc F \rightarrow \bigcirc \overline{F}) \tag{10}$$

for any fluent F and the axiom schemata:

$$F \vee \overline{F} \tag{11}$$
$$\Box(F \wedge \overline{F} \rightarrow \bot) \tag{12}$$
$$\Box(A \vee \neg A) \tag{13}$$
$$\Box(A_j \wedge A_k \rightarrow \bot) \tag{14}$$
$$\Box(A \rightarrow \mathit{someaction}) \tag{15}$$
$$\Box(\neg \mathit{someaction} \wedge \bigcirc \mathit{someaction} \rightarrow \bot) \tag{16}$$

for each $P \in \mathcal{F}$, and $A, A_j, A_k \in \mathcal{A}$, with $A_j \neq A_k$. Axiom (11) completes any fluent value at the initial situation by making it either true F or explicitly false $\overline{F}$. Axiom (12) rejects solutions where both F and $\overline{F}$ hold at the same state[3]. Axiom (13) gives freedom to choose any action occurrence at any state. This is then constrained by axiom (14), which avoids concurrence of two different actions, and Axioms(15),(16) that avoid performing new actions after a transition where no action was executed.

Given a $\mathcal{B}$ transition $\langle s, A, s' \rangle$ we say that a finite THT interpretation like $\langle \overline{T}, \overline{T} \rangle$ with $\overline{T} = \{T_0, T_1\}$, $T_0 = \{A\} \cup s$ and $T_1 = s'$ *corresponds* to $\langle s, A, s' \rangle$.

Theorem 1. *Given an action description D in $\mathcal{B}$, the set of transitions of D corresponds to the temporal equilibrium models of $\Pi(D) \cup \{\bigcirc \mathit{someaction} \rightarrow \bot\}$.*

Proof. Each formula $\Box\varphi$ can be instantiated as an infinite conjunction $\varphi \wedge \bigcirc\varphi \wedge \bigcirc^2\varphi \wedge \bigcirc^3\varphi \ldots$ In each $\bigcirc^i\varphi$ we may shift $\bigcirc$ using (3)-(6) to obtain Π':

$$\Pi(L \textbf{ if } \varphi) = \bigcirc^i\varphi \rightarrow \bigcirc^i L \tag{17}$$
$$\Pi(A \textbf{ causes } L \textbf{ if } \varphi) = \bigcirc^i\varphi \wedge \bigcirc^i A \rightarrow \bigcirc^{i+1} L \tag{18}$$
$$\bigcirc^i F \wedge \neg \bigcirc^{i+1} \overline{F} \rightarrow \bigcirc^{i+1} F \tag{19}$$
$$\bigcirc^i \overline{F} \wedge \neg \bigcirc^{i+1} F \rightarrow \bigcirc^{i+1} \overline{F} \tag{20}$$
$$\bigcirc^i F \vee \bigcirc^i \overline{F} \tag{21}$$
$$\bigcirc^i F \wedge \bigcirc^i \overline{F} \rightarrow \bot \tag{22}$$
$$\bigcirc^i A \vee \neg \bigcirc^i A \tag{23}$$
$$\bigcirc^i A_j \wedge \bigcirc^i A_k \rightarrow \bot \tag{24}$$
$$\bigcirc^i A \rightarrow \bigcirc^i \mathit{someaction} \tag{25}$$
$$\neg \bigcirc^i \mathit{someaction} \wedge \bigcirc^{i+1} \mathit{someaction} \rightarrow \bot \tag{26}$$

[3] Note that explicit negation has no logical entity in our current definition of TEL, and $\overline{F}$ is just interpreted as one more regular atom.

for $i = 0, 1, \ldots$ As the only modal operator is $\bigcirc$, Proposition 1 allows interpreting this theory under Equilibrium Logic for signature $\Sigma \cup \{\text{`}\bigcirc^i p\text{'} \mid p \in \Sigma, i = 0, 1, \ldots\}$ or, what is the same, consider the stable models of Π'. The resulting logic program is "temporally directed", i.e. given any rule r, for any atom $\bigcirc^i p$ in the head of r and any atom $\bigcirc^j q$ in its body, we have $j \leq i$. We can apply the splitting theorem [11] to gradually consider program fragments Π'_i consisting of the set of rules in Π' not containing atoms $\bigcirc^j p$ with $j > i$. The first of these programs, Π'_0, would contain rules (17),(21)-(25) reducing all expressions like $\bigcirc^i p = \bigcirc^0 p = p$. It is not difficult to see that the set of stable models of Π'_0 corresponds to $\{s \cup \{A\} \mid A \in \mathcal{A}$ and s is a complete state closed under $Z, A \in \mathcal{A}\}$, that is, all possible pairs $\langle s, A\rangle$. Now, consider any stable model T of Π'_1 so that $\langle s, A\rangle$ is stable model of Π'_0, whereas $s' = \{p \mid \bigcirc p \in T\}$. Once s and A are fixed, we can simplify $\Pi'_1 \setminus \Pi'_0$ so that rules (18) will amount to facts $\bigcirc L$ per each $L \in E(A, s)$, so $E(A, s) \subseteq s'$. On the other hand, if $F \in (s \cap s')$, we get $\bigcirc F \in T$ what together with (22) implies $\neg \bigcirc \overline{F}$, and so the antecedent of (19) holds, and we can replace this rule by the fact $\bigcirc F$. A similar reasoning can be done when $\overline{F} \in (s \cap s')$ to conclude that (20) can be replaced by fact $\bigcirc \overline{F}$. Thus, for a given $\langle s, A, s'\rangle$ we have a program containing a fact $\bigcirc L$ for any $L \in E(A, s) \cup (s \cap s')$. Since static laws Z are also present in Π_1 as formulas (17) with $i = 1$, it follows that $s' = Cn_Z(E(A, s) \cup (s \cap s'))$. Finally, axiom $\bigcirc someaction \rightarrow \bot$ avoids the inclusion of any $\bigcirc^i A$ for any action A and $i > 0$ and it can be easily seen that (19),(20) make the successor state s' persist for $i = 2, 3, \ldots,$ □

Theorem 2. *Given an action description D in $\mathcal{B}$, any temporal answer set of $\Pi(D)$ corresponds to a history of D.*

Proof. By induction on the length of the finite narrative n, taking as the base case Theorem 1 and repeating a similar reasoning.

5 Conclusions

In this paper we have provided a first definition for the linear temporal extension of Equilibrium Logic and gave some examples of its potential utility. As a result, we are able to capture the temporal component of planning problems inside a uniform logical formalism, being able to check properties like the non-existence of plan or the strong equivalence of different representations for the same transition system. On the other hand, the resulting logic may also offer interesting features for the typical application field of LTL, namely formal verification, as its nonmonotonicity allows expressing defaults, as for instance, the previously mentioned rule of inertia.

Much work remains to be done yet, both from a theoretical and a practical point of view. From the former, for instance, we can mention the axiomatisation for THT or a complexity assessment both for THT and TEL entailment. Another important issue is decidability, which seems easy to guess but has not been proved yet. From the practical point of view, obtaining inference methods (like semantic tableaux or temporal extensions of logic programming algorithms) or translating

other planning and action languages for their comparison are the two main lines for ongoing work.

Acknowledgements. We wish to thank David Pearce, Agustín Valverde, Manuel Ojeda and Alfredo Burrieza for an initial discussion motivating this work.

References

1. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R.A., Bowen, K.A. (eds.) Logic Programming: Proc. of the Fifth International Conference and Symposium, vol. 2, pp. 1070–1080. MIT Press, Cambridge, MA (1988)
2. Gelfond, M., Lifschitz, V.: Classical negation in logic programs and disjunctive databases. New Generation Computing 9, 365–385 (1991)
3. Pearce, D.: A new logical characterisation of stable models and answer sets. In: Dix, J., Przymusinski, T.C., Moniz Pereira, L. (eds.) NMELP 1996. LNCS, vol. 1216, Springer, Heidelberg (1997)
4. WASP: ASP solvers web page, last update (2005), `http://dit.unitn.it/~wasp/Solvers/index.html`
5. McCarthy, J., Hayes, P.: Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence Journal 4, 463–512 (1969)
6. Kamp, J.A.: Tense Logic and the Theory of Linear Order. PhD thesis, University of California at Los Angeles (1968)
7. Manna, Z., Pnueli, A.: The Temporal Logic of Reactive and Concurrent Systems: Specification. Springer, Heidelberg (1991)
8. Heyting, A.: Die formalen Regeln der intuitionistischen Logik. Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-mathematische Klasse, 42–56 (1930)
9. Gelfond, M., Lifschitz, V.: Action languages. Linköping Electronic Articles in Computer and Information Science 3(16) (1998)
10. Cabalar, P.: Temporal answer sets. In: Proceedings of the Joint Conference on Declarative Programming (AGP'99), pp. 351–365. MIT Press, L'Aquila, Italy (1999)
11. Lifschitz, V., Turner, H.: Splitting a logic program. In: International Conference on Logic Programming, pp. 23–37 (1994)

A Distributed Filesystem for Spare Storage*

Javier Paris, Victor M. Gulias, Alberto Valderruten, and Santiago Jorge

MADS Group. Department of Computer Science. University of A Coruna
{javierparis,gulias,valderruten,sjorge}@udc.es

Abstract. Having access to large quantities of storage space has always been a problem. Traditionally, large storage capacity has been expensive, using dedicated storage systems (NAS), or more complex networked storage (SAN). In many cases these solutions are overkill both in price and in features. Many small offices and research labs with limited budget have computers with unused disk space that could be used as shared storage.

A distributed filesystem can take advantage of the storage space of several computers to provide a larger storage. For a small office or lab this filesystem would have to provide easy integration into the infrastructure and reasonable scalability.

In this work we propose a filesystem that provides access to a large storage using the unused space in a network of workstations.

1 Introduction

A usual setup in a network of workstations in a small office or a small research laboratory is to have a centralized storage server which provides access through a network filesystem, such as NFS or SMB. This centralized storage server has traditionally been a server with a series of disk, but now can also be something more complex in structure like a SAN.

In this environment, the users store their files in the centralized storage, because that way they can access their files from anywhere. Usually it is also the only storage that is routinely backed up, which makes it more reliable than local storage.

However, most of the workstations have modern hard drives with large storage space which is not used because the centralized storage is more convenient. Due to this, there is usually a lot of unused space on the workstations, which typically only have the operating system and a few applications installed.

In many cases, the organization may have to update the storage server because the old centralized server runs out of space, while at the same time the disks in the workstations are almost empty. Even more, there may be some data in the centralized storage which the users keep there only because it is easier to access it from any workstation, but which is not so critical that it need to be backed up.

A distributed filesystem could provide access to this space, providing access to the disks in the workstations transparently. Ideally, it would do so without disturbing the normal usage of the systems.

* This work has been partially supported by MEC-TIN200508986 and Xunta de Galicia PGIDIT05TIC10503PR.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 249–256, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Most distributed filesystems are designed for clusters, with an emphasis on large scale storage and performance. However, in a network of workstation where the filesystem will be used mainly to store files not considered important enough for the centralized storage, performance and large scale are not that important. The main problem to be addressed in an scenario like this is the inherent unreliability of the used storage. Thus, distribution of all the information is critical, and there should be no single point of failure.

The design of this filesystem takes inspiration from the google filesystem[1]:

- The use of heavy duplication in unreliable storage. However, workstations are even more unreliable than the storage used in the google filesystem, but the data supposed to be stored in a filesystem like this is also less critical.
- The google filesystem takes away a lot of the complexity of metadata management by not having rewritable blocks. Blocks may be only written once. This is not feasible for a filesystem for general use like the one being proposed in this work. However, reads are optimized over writes.
- The google filesystem metadata service takes all its information from the blocks in the system. This idea is also used here, as it simplifies the management of the metadata nodes. The cost is having to keep several copies of each block for reliability, which decreases the available space.

In other similar designs, such as lustre[2], the metadata is much simpler by using only one active metadata server. This simplifies the metadata management. Bear in mind that the performance offered by a filesystem like lustre is much higher than a filesystem like the described on this work.

2 Design

As stated on the introduction, the main goal is to design a distributed filesystem that uses the spare storage present in a typical network of workstations(*NOWs*). In many *NOWs*, the main storage is kept in a centralized server, which provides storage to all the workstation through a network filesystem such as NFS or SMB. Because of this, the hard drives of the workstation are usually only used for the operating system. Users usually store all their data on the centralized server because it is much easier to access from any other place.

Given the average of a hard drive installed in a workstation today, a lot of space is not used just because it is not easily accessible. A filesystem which provides access to the unused storage in all the workstations of the network is likely to provide more space than the usual centralized server.

The main problems to be addressed in this scenario are:

- The filesystem should only use spare storage. The filesystem should not require reserved space in the machine, and only use the space available. It should also try not to fill the available space in any workstation, if at all possible, to leave the user of the workstation room to work normally.
- Workstations cannot be considered reliable. A workstation may be shutdown at any time. Anything stored may not be accessible at a later time. The

filesystem should replicate the data in more than one workstation, to decrease the possibility of having blocks which cannot be accessed.
Another problem posed by unreliable workstations is the problem of managing metadata. Metadata should be accessible from more than one point, just in case the workstation where it is fails. These two problems point towards a design as decentralized as possible.

On the other hand, there are also assumptions that can be made, which simplify the design. In most cases, in an environment like the one described, the users of the workstations will store their files on this filesystem, and those files are likely only going to be accessed by them. There will be cases where more than one user tries to modify the same file, but it can be assumed that this will not be a very common case.

The block size used in the filesystem is 8 MB. It is bigger than the block sizes of network filesystems like NFS. The main advantage of this approach is the reduced metadata that the system has to keep. This way the information about a file is much smaller, most files will only have one block. Even big files of a few hundred MB will only have a small number of blocks, less than 100. The access to the metadata servers is thus reduced, as the operations to access a block are done directly with the storage layer.

Having to read or write 8MB over the network in each operation has an obvious impact on performance. To alleviate this the storage layer allows the reads and writes of smaller parts of blocks, so the actual data exchanged over the network can be as small as needed.

A good approach is to try to optimize the filesystem for reads and reasonably usable for writes, even if concurrent writes are slower.

The final design has three layers (see Figure 1):

- An storage layer, which just store blocks. There will be one storage service for each workstation in the network.
- A metadata layer, which provides all the structure in the filesystem. This metadata layer can be replicated in several nodes. The services in this layer are provided by several different services.
- An access layer, which provides access to the clients in the network. Ideally, there can be more than one way to access the filesystem.

2.1 Storage Layer

The storage layer provides a service to read and write blocks in a workstation that is to provide storage access. These services are uses by both the metadata and access layer. Some of the goals for the filesystem have an influence in the design of this layer.

As it has already pointed out, the filesystem should not interfere with the normal usage of the workstations as much as possible. In particular, the storage layer should not use more space than what is needed at any given moment. A way of doing this is storing the blocks as normal files in the host's filesystem.

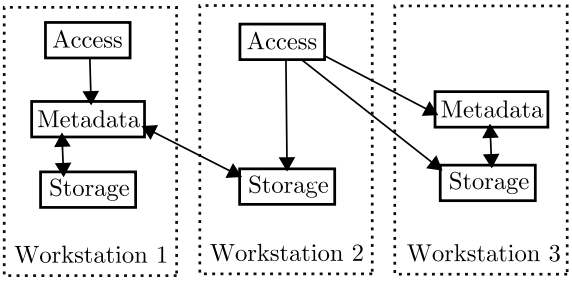


Fig. 1. Filesystem Layer Structure

This way only the space actually used by the blocks is not available to the user. Another advantage is that the filesystem may use up to the total free space in the workstation if the need arises. If the system used a separate partition the amount of space available to the filesystem would be fixed, and unusable for the user of the workstation.

The only potential disadvantage of storing blocks as files is that it may reduce performance, as any access to a block requires some system calls to navigate the filesystem and get to the file. However, this is not expected to have a very big impact on performance, as the block size chosen is big.

The filesystem identifies blocks using:

- A file Id, which identifies the file the block belongs to.
- A block number, which identifies the position of this block inside the file.
- A version, which is a number used to identify if a certain block is up to date.

This file identification is used then to name the files in which the blocks are stored. For example, the block number 3 in the file 300, in its first version would be the file "300-2-0". With this naming schema, the storage layer can know which blocks it stores any time it is started just by reading the list of files in the directory where the blocks are stored.

2.2 Metadata

The metadata layer keeps the information about the distributed filesystem, such as what the directory structure is, which files there are, or which the current version for each of its blocks is. There are several different services in this layer (Figure 2):

- A *Census* service, which is a service where all the different subsystems in the filesystem must register. This service is unique for all the filesystem, it runs only in one of the workstations. If for any reason it falls down, it is restarted in a different one, using the distribution system of Erlang[3]. This service makes it possible to have workstations joining and detaching from the system without creating problems. When a new storage node appears,

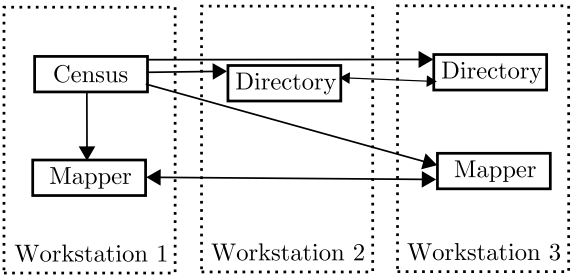


Fig. 2. Metadata Layer

it registers with this service, which get the blocks stored in that server. This information is then provided to a *mapper* service. If a node detaches from the system (either by deregistering, or by disappearing, for example in a power down) this services also notices a mapper so that the blocks stored in that node are deleted from the available block list.

– A *Mapper* service, which provides a localization service for the blocks in the files. Each block in a file is duplicated in more than one server, and the storage nodes may have more than one version at once. This services keeps track of all the blocks present in the currently accessible storage nodes. There may be more than one mapper in the system, and a distributed database(mnesia [4]) is used for synchronization between them.

It is important to notice that it only keeps information about the blocks which can be accessed. If a storage node becomes unavailable, its blocks are removed from the database. As there is no permanent information, this database is only kept in memory, which simplifies its management a lot. When the first mapper in the system is started, it initializes the database. When storage nodes register with the census, their block lists are added to the database. If more mappers appear at other nodes, they synchronize their database with the existing mappers. There is no main mapper, so this service is completely distributed.

Even if all the mappers fall down, the service is only interrupted until a new one is started, as this mapper will copy the block lists again from the storage nodes. Not having persistent information in the mappers makes this much more easy. The problem that arises from this design is the system cannot guarantee access to the most recent version of a block in the system. If the storage node has the current version for a block, but is down, the database may be initialized with an older version for that block in the file. In that case, information could be lost.

The mapper keeps track of the most recent versions of the blocks in files. Whenever there is a write to a block, the mapper creates a new block with the new block version. If at all possible, an old version of the block is overwritten in the process for efficiency.

- A *directory* service, which keeps information about the directory structure of the filesystem, and which files are stored inside it. This service also uses a distributed database for synchronization. The directory structure is stored using special blocks in the storage service. In this way, all information about the filesystem is stored in the storage nodes. As the blocks in the storage nodes are duplicated, the information should be accessible at any time.
- A *replica* service, which controls the copies of each block present on the system. Whenever the number of copies of a block get below a certain threshold, this service makes a copy of the block to another storage server. Similarly, when a block get obsolete because there is a new version, this service may trigger its overwrite by the new version, or deletes it.

2.3 Access Layer

The access layer provides access to the filesystem for usage in the clients. This layer uses the services provided by the metadata and storage layers to provide its services. It queries the directory service for accessing the structure of the filesystem. When it want to access the information in a file, it calls the mapper for this information. When it knows the current version of a block and its locations, it contacts the storage layer directly for the block.

The interface to the user can be provided in several ways. As an example, an NFS server implemented in Erlang was used to provide external access to this filesystem. In the future it would be desirable to have a native interface using tools like FUSE[5], which provide interfaces to implement filesystems in userspace.

3 Tests

Two test were performed in a prototype implementation. The system was setup with two storage nodes and two metadata servers. The nodes are a Athlon XP 1700+, with 512MB of RAM memory. The other is a Pentium 4 running at 2.8 Ghz, with 1GB of RAM memory. Both are connected using a 100Mbps switched network. A client launches requests to them.

The first test (Figure 3) reads blocks from the filesystem varying the block size of the read operation. As the filesystem adapts the block size of the internal operations to avoid working with 8MB blocks all the time, this has an effect on performance. As it can be seen, the performance increases until it reaches a maximum, which in this case is near the transfer capacity of the network.

In the write test (Figure 4), the behaviour is similar, but with some differences. The transfer rate is not so regular. This is due to the copy behaviour being triggered to keep the replication of the blocks the test is writing. Also the performance is lower, less than half the read performance, due to a higher usage of the network for the copies.

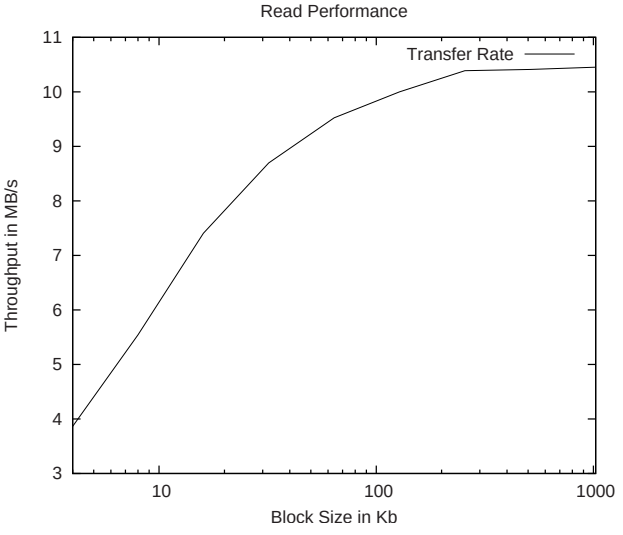


Fig. 3. Read Performance

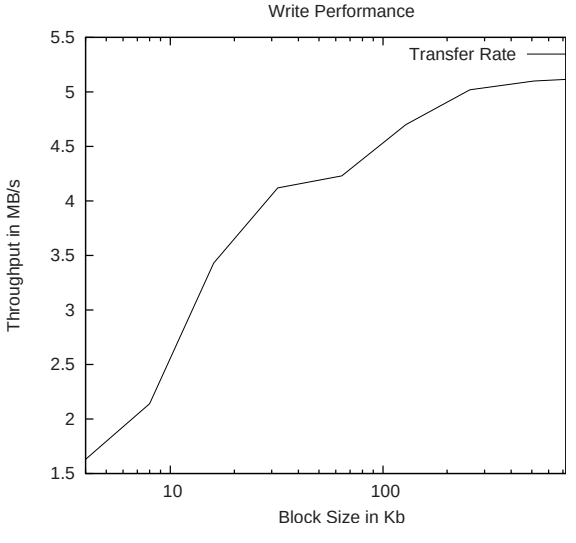


Fig. 4. Write Performance

4 Conclusions

In this work we have presented a filesystem that provides access to spare storage in a network of workstations. This environment favours a design that is as decentralized as possible. The problem in such a decentralized scenario is how

to keep the information coherent. In most distributed filesystems the metadata server is unique (although in many cases it can be replicated). Having the metadata information replicated in such a way that there is no central point in the system while at the same time having a modifiable filesystem which requires synchronization between all the metadata servers is the most difficult challenge in this design.

The filesystem presented has no single point of failure, as all the subsystems but the *Census* can be replicated. However ,the *Census* restarts in another node in the system in case of failure. The tradeoff is an increased usage of disk space, by keeping all the information replicated.

References

1. Ghemawat, S., Gobioff, H., Leung, S.T.: The google file system. In: SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles, pp. 29–43. ACM Press, New York (2003)
2. Schwan, P.: Lustre: Building a file system for 1000-node clusters (2003)
3. Armstrong, J., Virding, R., Wikström, C., Vlissides, J.: Concurrent programming in erlang, 2nd edn (1996)
4. Mattsson, H., Nilsson, H., Wikström, C.: Mnesia — A distributed robust DBMS for telecommunications applications. In: Gupta, G. (ed.) PADL 1999. LNCS, vol. 1551, pp. 152–163. Springer, Heidelberg (1999)
5. Szeredi, M.: Fuse: Filesystem in userspace (2005), `http://fuse.sourceforge.net/`

Generation of Indexes for Compiling Efficient Parsers from Formal Specifications

Carlos Gómez-Rodríguez[1], Miguel A. Alonso[1], and Manuel Vilares[2]

[1] Departamento de Computación, Universidade da Coruña
Facultad de Informtica, Campus de Elviña 5, 15071 La Coruña Spain
{cgomezr, alonso}@udc.es
[2] Departamento de Informática, Universidade de Vigo
E.T.S. de Ingeniería Informática, Campus As Lagoas, 32004 Orense Spain
vilares@uvigo.es
http://www.grupocole.org/

Abstract. Parsing schemata provide a formal, simple and uniform way to describe, analyze and compare different parsing algorithms. The notion of a parsing schema comes from considering parsing as a deduction process which generates intermediate results called *items*. An initial set of items is directly obtained from the input sentence, and the parsing process consists of the application of inference rules (called *deductive steps*) which produce new items from existing ones. Each item contains a piece of information about the sentence's structure, and a successful parsing process will produce at least one *final item* containing a full parse tree for the sentence or guaranteeing its existence. Their abstraction of low-level details makes parsing schemata useful to define parsers in a simple and straightforward way. Comparing parsers, or considering aspects such as their correction and completeness or their computational complexity, also becomes easier if we think in terms of schemata. However, when we want to actually use a parser by running it on a computer, we need to implement it in a programming language, so we have to abandon the high level of abstraction and worry about implementation details that were irrelevant at the schema level. In particular, we study in this article how the source parsing schema should be analysed to decide what kind of indexes need to be generated in order to obtain an efficient parser.

1 Introduction

Parsing schemata are explained in detail in [14,11]. In this article, we will give a brief insight into the concept by introducing a concrete example: a parsing schema for Earley's algorithm [3]. Given a context-free grammar $G = (N, \Sigma, P, S)$, where N denotes the set of nonterminal symbols, Σ the set of terminal symbols, P the production rules and S the axiom of the grammar, and a sentence of length n denoted by $a_1\ a_2\ \ldots\ a_n$, the schema describing Earley's algorithm is as follows:[1]

[1] From now on, we will follow the usual conventions by which nonterminal symbols are represented by uppercase letters ($A, B \ldots$), terminals by lowercase letters ($a, b \ldots$) and strings of symbols (both terminals and nonterminals) by Greek letters ($\alpha, \beta \ldots$).

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 257–264, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Item set:
$\{[A \to \alpha.\beta, i, j] \mid A \to \alpha\beta \in P \wedge 0 \leq i < j\}$

Initial items (hypotheses):
$\{[a_i, i-1, i] \mid 0 < i \leq n\}$

Deductive steps:

EARLEY INITTER: $\dfrac{}{[S \to .\alpha, 0, 0]}\ S \to \alpha \in P$

EARLEY SCANNER: $\dfrac{[A \to \alpha.a\beta, i, j] \qquad [a, j, j+1]}{[A \to \alpha a.\beta, i, j+1]}$

EARLEY PREDICTOR: $\dfrac{[A \to \alpha.B\beta, i, j]}{[B \to .\gamma, j, j]}\ B \to \gamma \in P$

EARLEY COMPLETER: $\dfrac{[A \to \alpha.B\beta, i, j] \qquad [B \to \gamma., j, k]}{[A \to \alpha B.\beta, i, k]}$

Final items:
$\{[S \to \gamma., 0, n]\}$

Items in the Earley algorithm are of the form $[A \to \alpha.\beta, i, j]$, where $A \to \alpha.\beta$ is a grammar rule with a special symbol (dot) added at some position in its right-hand side, and i, j are integer numbers denoting positions in the input string. The meaning of such an item can be interpreted as: "There exists a valid parse tree with root A, such that the direct children of A are the symbols in the string $\alpha\beta$, and the leaf nodes of the subtrees rooted at the symbols in α form the substring $a_{i+1} \ldots a_j$ of the input string".

The algorithm will produce a valid parse for the input sentence if an item of the form $[S \to \gamma., 0, n]$ is generated: according to the aforesaid interpretation, this item guarantees the existence of a parse tree with root S whose leaves are $a_1 \ldots a_n$, that is, a complete parse tree for the sentence.

A deductive step $\frac{\eta_1 \ldots \eta_m}{\xi}\ \Phi$ allows us to infer the item specified by its consequent ξ from those in its antecedents $\eta_1 \ldots \eta_m$. *Side conditions* (Φ) specify the valid values for the variables appearing in the antecedents and consequent, and may refer to grammar rules as in this example or specify other constraints that must be verified in order to infer the consequent.

1.1 Compilation of Parsing Schemata

Parsing schemata are located at a higher abstraction level than algorithms. As can be seen in the text above, a schema specifies the steps that must be executed and the intermediate results that must be obtained in order to parse a given

string, but it makes no claim about the order in which to execute the steps or the data structures to use for storing the results. To overcome this, we have developed a compiler that given a parsing schema is able to obtain an executable implementation of the corresponding parser [7,6]. The input to the compiler is a simple and declarative representation of a parsing schema, which is practically equal to the formal notation that we used previously. For example, a valid schema file describing the Earley parser would be:

```
@goal [ S -> alpha . , 0 , length ]

@step EarleyInitter
------------------------ S -> alpha
[ S -> . alpha , 0 , 0 ]

@step EarleyScanner
[ A -> alpha . a beta , i , j ]
[ a , j , j+1 ]
---------------------------------
[ A -> alpha a . beta , i , j+1 ]
```

```
@step EarleyCompleter
[ A -> alpha . B beta , i , j ]
[ B -> gamma . , j , k ]
---------------------------------
[ A -> alpha B . beta , i , k ]

@step EarleyPredictor
[ A -> alpha . B beta , i , j ]
-------------------------- B -> gamma
[ B -> . gamma , j , j ]
```

1.2 Related Work

Shieber *et al.* provide in [12] a Prolog implementation of a deductive parsing engine which can also be used to implement parsing schemata. However, its input notation is less declarative (since schemata have to be programmed in Prolog) and it does not support automatic indexing, so the resulting parsers are inefficient unless the user programs indexing code by hand, abandoning the high abstraction level.

Another alternative for implementing parsing schemata, the Dyna language [4], can be used to implement some kinds of dynamic programs; but it has a more complex and less declarative notation than ours, which is specifically designed for denoting schemata.

2 Automatic Generation of Indexes for Parsing Schemata

In this section, we study how the compiler analyses the source parsing schema to decide what kind of indexes need to be generated in order to obtain an efficient parser. In particular, implementation of the following operations affects the resulting parser's computational complexity:

- Check if a given item exists in the item set.
- Search the item set for all items satisfying a certain specification: Once an antecedent of a deductive step has been matched, a search for items matching the rest of the antecedents is needed in order to make inferences using that step.

In order to maintain the theoretical complexity of parsing schemata, we must provide constant-time access to items. In this case, each single deduction takes

place in constant time, and the worst-case complexity is bounded by the maximum possible number of step executions: all complexity in the generated implementation is inherent to the schema.

In order to achieve this, we propose to generate two distinct kind of indexes for each schema, corresponding to the operations mentioned before:

- Items are indexed in *existence indexes*, used to check whether an item exists in the item set, and *search indexes*, which allow us to search for items conforming to a given specification.
- Deductive steps are indexed in *deductive step indexes*. These indexes are used to restrict the set of "applicable deductive steps" for a given item, discarding those known not to match it. Deductive step indexes usually have no influence on computational complexity with respect to input string size, but they do have an influence on complexity with respect to the size of the grammar, since the number of step instances depends on grammar size when grammar productions are used as side conditions.

2.1 A Simple Case

The generation of indexing code is not trivial, since the elements by which we should index items in order to achieve efficiency vary among schemata. For instance, if we are trying an Earley COMPLETER step

$$\frac{[A \to \alpha.B\beta, i, j], [B \to \gamma., j, k]}{[A \to \alpha B.\beta, i, k]}$$

on an item of the form $[B \to \gamma., j, k]$ which matches the second antecedent, we will need to search for items of the form $[A \to \alpha.B\beta, i, j]$, for any values of A, α, β and i, in order to draw all the possible conclusions from the item and step. Since the values of B and j are fixed, this search will be efficient and provide constant-time access to items if we have them indexed by the symbol that follows the dot and by the second string position (B and j). However, if we analyze the other Earley steps in the same way, we will find that their indexing needs are different, and different parsing schemata will obviously have different needs.

Therefore, in order to generate indexing code, we must take the distinct features of each schema into account. In the case of search indexes, we must analyze each deductive step just as we have analyzed the COMPLETER step: we must keep track of which variables are instantiated to a concrete value when a search must be performed. This information is known at schema compilation time and allows us to create indexes by the elements corresponding to instantiated variables. For example, in the case of COMPLETER, we would create the index that we mentioned before (by the symbol directly after the dot and the second string position) and another index by the symbol in the left side of productions and the first string position. This second index is useful when we have an item matching the first antecedent and we want to search for items matching the second one, and is obtained by checking which variables are instantiated when the first antecedent is matched.

The generation of existence indexes is similar to, but simpler than, that of search indexes. The same principle of checking which variables will be instantiated when the index is needed is valid in this case, but when an item is checked for existence it is always fully known, so all its variables are instantiated.

Deductive step indexes are generated by taking into account those step variables which will take a value during instantiation, i.e. which variables appear on side conditions. Since these variables will have a concrete value for each instance of the step, they can be used to filter instances in which they take a value that will not allow matching with a given item.

2.2 A General Case

For a more general view of how the adequate indexes can be determined by a static analysis of the schema prior to compilation, we can analyze the general case where we have a deductive step of the form

$$\frac{[a, d, e, g], [b, d, f, g]}{consequent} \quad c\ e\ f\ g$$

where each lowercase letter represents the set of elements (be them grammar symbols, string positions or other entities) appearing at particular positions in the step, so that a stands for the set of elements appearing only in the first antecedent item, e represents those appearing in the first antecedent and side condition, g those appearing in both antecedents and side condition, and the rest of the letters represent the other possible combinations as can be seen in the step. We can easily see that any deductive step with two antecedent items can be represented in this way (note that the element sets can be empty, and that the ordering of elements inside items is irrelevant to this discussion).

In this case, the following indexes are generated:

- One deductive step index for each antecedent, using as keys the elements appearing both in the side condition *and* in that particular antecedent: therefore, two indexes are generated using the values (e, g) and (f, g). These indexes are used to restrict the set of deductive step instances applicable to items. As each instance corresponds to a particular instantiation of the side conditions, in this case each step instance will have different values for c, e, f and g. When the deductive engine asks for the set of steps applicable to a given item $[w, x, y, z]$, the deductive step handler will use the values of (y, z) as keys in order to return only instances with matching values of (e, g) or (f, g). Instances of the steps where these values do not match can be safely discarded, as we know that our item will not match any of both antecedents.
- One search index for each antecedent, using as keys the elements appearing in that antecedent which are also present in the side condition *or* in the other antecedent. Therefore, a search index is generated by using (d, e, g) as keys in order to recover items of the form $[a, d, e, g]$ when d, e and g are known and a can take any value; and another index using the keys (d, f, g)

is generated and used to recover items of the form $[b, d, f, g]$ when d, f and g are known. The first index allows us to efficiently search for items matching the first antecedent when we have already found a match for the second, while the second one can be used to search for items matching the second antecedent when we have started our deduction by matching the first one.
- One existence index using as keys all the elements appearing in the consequent, since all of them are instantiated to concrete values when the step successfully generates a consequent item. This index is used to check whether the generated item already exists in the item set before adding it.

As this index generation process must be applied to all deductive steps in the schema, the number of indexes needed to guarantee constant-time access to items increases linearly with the number of steps. However, in practice we do not usually need to generate all of these indexes, since many of them are repeated or redundant. For example, if we suppose that the sets e and f in our last example contain the same number and type of elements, and elements are ordered in the same way in both antecedents, the two search indexes generated would in fact be the same, and our compiler would detect this fact and generate only one. In practical cases, the items used by different steps of a parsing schema usually have the same structure, so most indexes can be shared among several deductive steps and the amount of indexes generated is small.

In our example, we have considered only two antecedents for the sake of simplicity, but the technique is general and can be applied to deductive steps with an arbitrary number of antecedents.

3 Experimental Results

We have used our technique to generate implementations of three popular parsing algorithms for context-free grammars: CYK [8,15], Earley and Left-Corner [9]. The three algorithms have been tested with sentences from three different natural language grammars: the English grammar from the Susanne corpus [10], the Alvey grammar [2] (which is also an English-language grammar) and the Deltra grammar [13], which generates a fragment of Dutch. The Alvey and Deltra grammars were converted to plain context-free grammars by removing their arguments and feature structures. The test sentences were randomly generated.[2] Performance results for all these algorithms and grammars are shown in Table 1.[3] The success of the index generation technique proposed in this article is shown by the fact that the empirical computational complexity of the three parsers is

[2] Note that, as we are interested in measuring and comparing the performance of the parsers, not the coverage of the grammars; randomly-generated sentences are a good input in this case: by generating several sentences of a given length, parsing them and averaging the resulting runtimes, we get a good idea of the performance of the parsers for sentences of that length.

[3] Tests performed on a laptop with an Intel 1500 MHz Pentium M processor, 512 MB RAM, Sun Java Hotspot virtual machine (version 1.4.2_01-b06) and Windows XP.

Table 1. Performance measurements for generated parsers

Grammar	String length	Time Elapsed (s)			Items Generated		
		CYK	Earley	LC	CYK	Earley	LC
Susanne	2	0.000	1.450	0.030	28	14,670	330
	4	0.004	1.488	0.060	59	20,945	617
	8	0.018	4.127	0.453	341	51,536	2,962
	16	0.050	13.162	0.615	1,439	137,128	7,641
	32	0.072	17.913	0.927	1,938	217,467	9,628
	64	0.172	35.026	2.304	4,513	394,862	23,393
	128	0.557	95.397	4.679	17,164	892,941	52,803
Alvey	2	0.000	0.042	0.002	61	1,660	273
	4	0.002	0.112	0.016	251	3,063	455
	8	0.010	0.363	0.052	915	7,983	1,636
	16	0.098	1.502	0.420	4,766	18,639	6,233
	32	0.789	9.690	3.998	33,335	66,716	39,099
	64	5.025	44.174	21.773	133,884	233,766	170,588
	128	28.533	146.562	75.819	531,536	596,108	495,966
Deltra	2	0.000	0.084	0.158	1,290	1,847	1,161
	4	0.012	0.208	0.359	2,783	3,957	2,566
	8	0.052	0.583	0.839	6,645	9,137	6,072
	16	0.204	2.498	2.572	20,791	28,369	22,354
	32	0.718	6.834	6.095	57,689	68,890	55,658
	64	2.838	31.958	29.853	207,745	282,393	261,649
	128	14.532	157.172	143.730	878,964	1,154,710	1,110,629

below their theoretical worst-case complexity of $O(n^3)$. Similar results have been obtained for highly ambiguous artificial grammars such as the one used in [1].

4 Conclusions

Parsing algorithms can be defined in a simple, formal and uniform way by means of Parsing Schemata, and there exists a compilation technique which allows us to automatically transform a parsing schema into an implementation of the algorithm it describes. In this article we have shown how adapted indexing code can be automatically generated for parsing schemata, so that the generated parsers keep the theoretical computational complexity of the parsing algorithms.

Although all the cases we have seen so far correspond to context-free parsing, our technique is not limited to working with context-free grammars, since parsing schemata can be used to represent parsers for other grammar formalisms as well. All grammars in the Chomsky hierarchy can be handled in the same way as context-free grammars. For example, we have generated implementations for some of the most popular parsers for Tree Adjoining Grammars [5,6].

Acknowledgements

The work reported in this article has been supported in part by Ministerio de Educación y Ciencia (MEC) and FEDER (TIN2004-07246-C03-01, TIN2004-07246-C03-02), Xunta de Galicia (PGIDIT05PXIC30501PN, PGIDIT05PXIC10501PN, Rede Galega de Procesamento da Linguaxe e Recuperación de Información) and Programa de Becas FPU of MEC.

References

1. Alonso, M.A., Cabrero, D., Vilares, M.: Construction of Efficient Generalized LR Parsers. In: Wood, D., Yu, S. (eds.) WIA 1997. LNCS, vol. 1436, pp. 7–24. Springer, Heidelberg (1998)
2. Carroll, J.A.: Practical unification-based parsing of natural language. TR no. 314, University of Cambridge, Computer Laboratory, England. PhD Thesis (1993)
3. Earley, J.: An efficient context-free parsing algorithm. Communications of the ACM 13(2), 94–102 (1970)
4. Eisner, J., Goldlust, E., Smith, N.A.: Dyna: A declarative language for implementing dynamic programs. In: Proceedings of ACL 2004 (Companion Volume), July 2004, pp. 218–221. Barcelona (2004)
5. Gómez-Rodríguez, C., Alonso, M.A., Vilares, M.: On theoretical and practical complexity of TAG parsers. In: Monachesi, P., Penn, G., Satta, G., Wintner, S. (eds.) FG 2006: The 11th conference on Formal Grammar. Center for the Study of Language and Information, Malaga, Spain, July 29-30, vol. 5, pp. 61–75. Stanford (2006)
6. Gómez-Rodríguez, C., Alonso, M.A., Vilares, M.: Generating XTAG parsers from algebraic specifications. In: Proceedings of the 8th International Workshop on Tree Adjoining Grammar and Related Formalisms. Sydney, July 2006, pp. 103-108, Association for Computational Linguistics, East Stroudsburg, PA (2006)
7. Gómez-Rodríguez, C., Vilares, J., Alonso, M.A.: Automatic Generation of Natural Language Parsers from Declarative Specifications. In: L. Penserini, P. Peppas and A. Perini (eds.), STAIRS 2006 - Proceedings of the Third Starting AI Researchers' Symposium, Riva del Garda, Italy, August 2006, vol. 142 of Frontiers in Artificial Intelligence and Applications, pp. 259-260, IOS Press, Amsterdam (2006)
8. Kasami, T.: An efficient recognition and syntax algorithm for context-free languages. Scientific Report AFCRL-65-758, Air Force Cambridge Research Lab, Bedford, Massachussetts (1965)
9. Rosenkrantz, D.J., Lewis II, P.M.: Deterministic Left Corner parsing. In: Conference Record of 1970 Eleventh Annual Meeting on Switching and Automata Theory, pp. 139–152. IEEE Computer Society Press, Los Alamitos (1970)
10. Sampson, G.: The Susanne corpus, Release 3 (1994)
11. Schneider, K.-M.: Algebraic Construction of Parsing Schemata. Mensch & Buch Verlag, Berlin, Germany (2000)
12. Shieber, S.M., Schabes, Y., Pereira, F.C.N.: Principles and implementation of deductive parsing. Journal of Logic Programming 24(1–2), 3–36 (1995)
13. Schoorl, J.J., Belder, S.: Computational linguistics at Delft: A status report, Report WTM/TT 90–09 (1990)
14. Sikkel, K.: Parsing Schemata — A Framework for Specification and Analysis of Parsing Algorithms. Springer, Berlin (1997)
15. Younger, D.H.: Recognition and parsing of context-free languages in time n^3. Information and Control 10(2), 189–208 (1967)

From Text to Knowledge*

M. Fernández[1], E. Villemonte de la Clergerie[2], and M. Vilares[1]

[1] Department of Computer Science, University of Vigo
Campus As Lagoas s/n, 32004 Ourense, Spain
{mfgavilanes,vilares}@uvigo.es
[2] Institut National de Recherche en Informatique et en Automatique
Domaine de Voluceau, Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France
Eric.De_La_Clergerie@inria.fr

Abstract. In this paper, we present a new approximation in Natural Language Processing (NLP) aimed at knowledge representation and acquisition using a formal syntactic frame. In practice, we introduce our implementation on an encyclopedic corpus in a botanic domain, illustrating the algorithm on a set of preliminary tests.

1 Introduction

Many documents written and published before the computer age are documents with an important content which is hard to search and utilize, because of the lack of automatic tools. To make the available information accessible in textual format, it is crucial to capture relevant data and convert them into a formal representation that will be used to help users. To do so, it is not feasible to recover the logical structure manually and, depending on the documents, it is a difficult task to consider automatic analyzers. This justifies the recent interest in automatic knowledge acquisition and, in particular, into applications on specific practical domains.

There are two main approaches in order to deal with the acquisition of semantic relations between terms. On the one hand, methods based on the comparison of syntactic contexts, often based on the use of statistic models. Briefly, some relations between terms are characterized by specific constructions that we can locate in texts from a particular lexical/syntactical pattern. Later, by using statistical calculus to compute the number of occurrences of the terms in these documents, we will be able to create a graph so as to extract specific collocations. The graph of co-occurrences are above all used to discover the sense and different applications of words through the detection of relevant structures in the graph, such as close noun sentences [1].

Other methods are grammar-oriented in the sense that they work directly on a parsing process, which serves as guideline for a more sophisticated linguistic

* Research partially supported by the Spanish Government under project TIN2004-07246-C03-01, and the Autonomous Government of Galicia under project PGIDIT05PXIC30501PN and the Network for Language Processing and Information Retrieval.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 265–272, 2007.
© Springer-Verlag Berlin Heidelberg 2007

analysis. Texts are parsed in order to locate relations that can allow classes of words to be formed. In this way, the aim is to find similar terms that can later be grouped and classified in a graph. In this sense, some authors suggest conflating candidates that are variants of each other through a self-indexing procedure [4], while others [2] propose post-process parses so as to emphasize relationships between words.

We combine these two approaches in a proposal including original contributions. The acquisition phase is performed from a shallow parser, whose kernel is a *tree-adjoining grammar* (TAG) [5], a mildly context-sensitive formalism that improves parse recognition power in comparison to classic context-free grammars. The resulting partial parsing structures introduce ambiguities that we solve on the basis of an error-mining strategy [7]. Later, on the linguistic side, the French grammar used is compiled from a source *meta-grammar* (MG) [9], which is modular and hierarchically organized.

2 The Running Corpus

This research was conducted using a botanic corpus describing West African flora. We concentrate on the *"Flore du Cameroun"* published between 1963 and 2001, which includes about forty volumes in French, each volume running to about three hundred pages, organized as a sequence of sections, each one following a systematic structural schema and relating different species. So, sections are divided into a set of paragraphs enumerating morphological aspects such as color, size or form. This involves a nominal regime in sentences, named entities to denote dimensions, and also adverbs which modify the meaning of the verbs or adjectives so as to express frequency and intensity. The corpus[1] compiles typical vocabulary for the majority of the text based on this matter, and we consider it to be sufficiently representative for our purposes.

Our work forms part of the BIOTIM[2] [6] project on processing botanical corpora. We omit the initial phases of the project, related to the transfer from textual to electronic format [7] by means of an OCR, and also the capture of the logical structure of the text to browse it, through the combination of mark-up language and chunking tasks.

3 The Parsing Frame

Shallow parsing techniques for information extraction often rely on hand-crafted extracting patterns, a costly task that has to be redone for each new domain, that introduces ambiguities due to the high frequency of unknown terms.

In this context, the parsing frame we choose to work with is DyALog [8], taking TAG as grammatical formalism. We justify this on the basis that its dynamic programming architecture allows us to benefit from sharing of

[1] Provided by the French Institute of Research for Cooperative Development.
[2] http://www-rocq.inria.fr/imedia/biotim/

parse computations and representations in dealing with non-determinism which improves efficiency. On the other hand, the consideration of TAG as grammatical formalism powers the system with new linguistic capabilities as, for example, cross and nested references as well as the constant growth property[3]. We do that by saving the valid prefix[4] and constant growth[5] properties and the polynomial complexity from context-free language. DyALog returns a total or partial parsing shared-forest on the basis of a TAG of large coverage for French. In this sense, the parser was improved, tailoring *meta-grammar* [6] and using error-mining techniques to track the words that occur more often than expected in unparsable sentences.

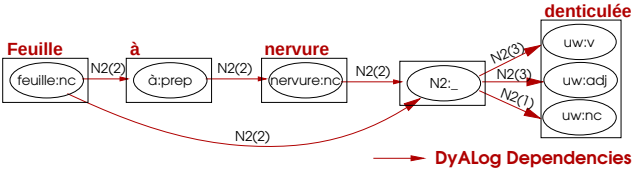


Fig. 1. Shared-parse dependencies from DyAlog

In Fig. 1, we illustrate the parse output for the sentence *"feuille à nervure denticulée"*, from now on our running example. The rectangular shapes represent *clusters*, that is, forms that refer to a position in the input string and all the possible lemmas with their corresponding lexical categories. We call these *nodes*, represented by ellipses. So, lexical ambiguities correspond to clusters containing nodes with different lemmas, or the same lemma associated to different lexical categories. Finally, arrows represent binary dependencies between words through some syntactic construction, showing syntactic ambiguities. So, the parse provides the mechanisms to deal with a posterior semantic phase of analysis, avoiding the elimination of syntactic data until we are sure it is unnecessary for knowledge acquisition.

3.1 Lexical Ambiguities

The morpho-syntactic analysis of the sentences is performed by concatenating a number of different tasks compliant with the MAF proposal[7]. Its description is not the goal of this work, and we shall only focus on the operational purpose.

In effect, in the context described, tagging becomes a non-deterministic and incomplete task, even if the vocabulary is relatively restricted such as in this

[3] It makes reference to linguistic intuition that sentences in a natural language can be built from a finite set of enclosed constructions by means of lineal operations.
[4] It guarantees that, as they read the input strings from left-to-right, the sub-strings read so far are valid prefixes for the language.
[5] It establishes the independence of each adjunction in a TAG with respect to others.
[6] Which is modular and hierarchically organized
[7] http://atoll.inria.fr/catalogue.en.html#Lingua::MAF

encyclopedic corpus. Most of the common words used in these documents are nouns and adjectives which do not appear in the lexicon or dictionaries, causing them to be considered as unknown words. So, in our running example of Fig. 1, the word *"denticulée"* (`"dentate"`) has been assigned with a label unknown (`uw`), and several lexical categories are in competition following a grammar oriented approach. In fact, there are three possible associated lexical categories: verb (`v`), adjective (`adj`) and noun (`nc`). These ambiguities cannot always be solved at lexical level and should be left to be considered at parsing time, introducing an additional factor of syntactic ambiguity.

3.2 Syntactic Ambiguities

Parsing in NLP with shallow/partial strategies often translates into only capturing local syntactic phenomena. This is an important drawback, because much of our information is expressed as nominal sentences, where the main noun is followed by prepositional attachments, as in the sentence *"feuille à nervure denticulée"*, in which we could locally consider two different interpretations: `"leaf with dentate vein"` or `"dentate leaf with vein"`. It becomes impossible here to establish if the word *"denticulée"*(`"dentate"`) relates to *"feuille"* (`"leaf"`) or to *"nervure"*(`"vein"`), as is shown in Fig. 1.

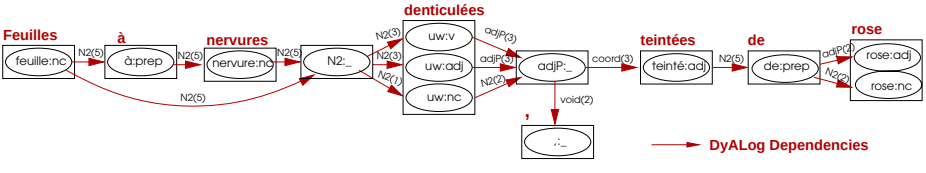


Fig. 2. Shared-parse dependencies from DyAlog

Another source of ambiguities is when the noun is followed by an enumeration of properties, i.e. coordination structures relating properties to a list of nouns, as in *"feuilles à nervures denticulées, teintées de rose"* (`"dentate leaf with vein, rose-tinted"`), the example shown in Fig. 2. Moreover, some of these properties may recursively be expressed as propositional sentences following the same structure. Therefore, we face non-determinism because of the attachment of a property to the main noun. For this reason, the above mentioned issues must be solved by relying on the use of a deep parser. Whichever the case is, the solution will lead us to use tools enabling us to go more deeply into the knowledge contained in the document.

4 Knowledge Acquisition

Once we recover the shared-forest of dependencies for each total/partial parse generated by DyALog, we try to effectively extract the latent semantics in the

document. In essence the idea is, by means of additional information from the corpus, to detect and eliminate useless dependencies. So, the lexical ambiguity shown in Fig. 1 should be decided in favor of the second translation `"dentate leaf with vein"`, because the word *"denticulée"* (`"dentate"`) is a typical characteristic that could only be applied on the word *"feuille"* (`"leaf"`) and not on the word *"nervure"* (`"vein"`). Furthermore, thanks to the encyclopedic corpus on botany, we could confirm this by examining it thoroughly. In this sense, to solve the ambiguity, we are looking for the information considering an iterative learning process to achieve our aim, in which term extraction constitutes the starting phase to formalize such a task.

Table 1. Learning the lexical category of *"denticulée"*

1. $P(\text{denticulée, adj})^{\text{cluster(j)}}_{\text{loc(0)}} = \dfrac{P(\text{denticulée, adj})^{\text{cluster(j)}}_{\text{loc}} * \#\text{arc}}{\Sigma_X P(\text{denticulée, X})^{\text{cluster(j)}}_{\text{loc}} * \#\text{arc}}$

2. $P(\text{denticulée, adj})_{\text{glob(n+1)}} = \dfrac{\Sigma_{j=1}^{m} P(\text{denticulée, adj})^{\text{cluster(j)}}_{\text{loc(n)}}}{\#\text{occ}}$

3. $P(\text{denticulée, adj})^{\text{cluster(j)}}_{\text{loc(n+1)}} = \dfrac{P(\text{denticulée, adj})^{\text{cluster(j)}}_{\text{loc(n)}} * P(\text{denticulée, adj})_{\text{glob(n+1)}}}{\Sigma_X \ P(\text{denticulée, X})^{\text{cluster(j)}}_{\text{loc(n)}} * P(\text{denticulée, X})_{\text{glob(n+1)}}}$

4.1 Term Extraction

Before running the term extraction module, it is necessary to perform a previous step. The vocabulary used is quite limited. So, it is quite frequent to find words in the corpus which do not appear in the lexicon (unknown words). For those words many lexical categories could be suggested. Therefore, this step relies in adapting the error-mining principle to be able to identify the correct one, a protocol that can be also applied if we need to identify the most probable lexical category in polysemous known words. In these documents words tend to be monosemous, which enables us to have an idea of the most probable lexical category, but does not discard other alternatives.

Focusing on the word *"denticulée"* (`"dentate"`) in Fig. 1, we try to identify the correct interpretation by introducing an iterative process to compute the probability associated to each alternative in lexical category assignment. Following items in Table 1, we have the estimation of this probability associated to the adjective category. In Table 2 we can see different examples and the probabilities for all of these alternatives after applying 25 iterations. More in detail, we introduce this process as follows:

1. We compute the local probability of the lexical category of each word in sentences. To start the process, we take into account the weight of the initial probability of the lexical category considered, denoted by $\mathtt{P}_{\text{loc}}$, that is a simple

ratio on all possible lexical categories of a cluster in a sentence. We also stake into account the number of input derivations for each node with different lexical categories in a cluster, denoted by *#arc*. The normalization is given by the possible lexical categories involving the cluster in the sentence and represented by variable `X`.

2. Then, we re-introduce the local probabilities into the whole corpus in each cluster, in order to re-compute the weights of all lexical categories, estimating then globally the most probable ones. The normalization is given by the number of occurrences of this word with this lexical category, `#occ`.
3. The local value in the new iteration should take into account both the global preferences and the local injection of these preferences in the sentences, reinforcing the local probabilities. The normalization is given by previous local and global weights of the lexical category represented by variable `X`.

After a number of iterations, a fixed point assures the convergence of the strategy, as shown in Table 2, illustrating the global probabilities in the corpus obtained for the words *"denticulée"* (`"dentate"`), *"nervilles"* (`"little veins"`), *"obtusément"* (`"obtusely"`) and *"viscidie"* (`"higher part of the soldered androecium and gynoecium"`).

Table 2. Global lexical categories obtained after 25 iterations

Form-lemma	**Possible lex. cat.**	**Probabilities**
denticulée - `uw`	adj	adj=1
(`"dentate"`)	nc, v	nc=0; v=0
	nc	nc=0.94
nervilles - `uw`	adv	adv=0.04
(`"little veins"`)	adj	adj=0.04
	np, v	np=0; v=0
obtusément - `uw`	adv	adv=1
(`"obtusely"`)	nc, adj, v	nc=0; adj=0; v=0
viscidie - `uw`	nc	nc=0.92
(`"higher part of the soldered`	v	v=0.08
`androecium and gynoecium"`)	np, adj, adv	np=0; adj=0; adv=0

Once we have computed the weights of the lexical categories for each word in the corpus, the next step is to generate new dependencies. For each parsed sentence, we identify generic lexical and/or syntactic patterns, through the existing shared-forest of dependencies. We look now for pairs referring to governor/governed relationships between words, from those represented by arrows in Figs. 1 and 2. In particular, because of the nominal regime, we are more interested in dependencies between nouns and adjectives. This justifies filtering those dependencies, as shown in Fig. 3, following the dotted lines. So, the word *"nervures"* (`"veins"`) is connected to *"denticulées"* (`"dentate"`), considering it as an adjective. Furthermore, we are also interested in extracting dependencies between nouns through, for example, prepositions such as *"feuilles à nervures"* (`"leaves with veins"`). In many cases the second part, that is the governed word, is a subclass of the first one, that is the governor word. The

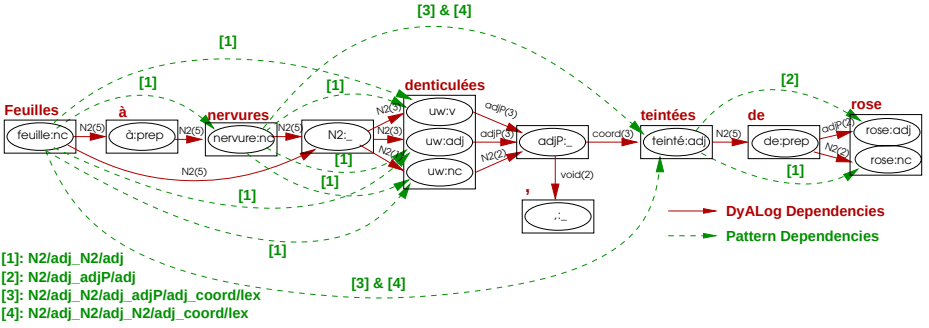


Fig. 3. New dependencies

same occurs with dependencies between adjectives such as *"teintées de rose"* (`("rose-tinted")`, *"rose"* being an adjective.

The acquisition of semantic classes will be done on these new dependencies, since full cascading is not currently implemented because of its complexity and computational cost. The new dependencies extracted are either filtered or chains of dependencies that are completed by additional direct dependencies, for instance chains of coordination or enumeration. For example, in DyALog dependencies, the word *"denticulées"* (`("dentate")`) is connected with *"teintées"* (`("tinted")`) through an enumeration. Using our term extraction module, we can connect the word *"teintées"* (`("tinted")`) to other governor words such as *"feuilles"* (`("leaves")`) or *"nervures"* (`("veins")`). Furthermore, we also filtered out some that we consider not very pertinent, such as adverbial modifiers. Now we are able to introduce our iterative knowledge acquisition process.

4.2 The Iterative Process

Although robust parsing makes it possible to detect syntactic patterns following the distributional hypothesis [3], we need to simplify the graph of dependencies resulting from the previous step in order to detect pertinent relations by trying to delete the spurious interpretations for the domain. In this sense, we establish a simple syntactic constraint: a governed word can only have one governor. It is sensible to think that a word can be related to only one, such as in Fig. 3, where *"teintées"* (`("tinted")`) could be governed by *"feuilles"* (`("leaves")`) or by *"nervures"* (`("veins")`) and, in consequence, we should eliminate one of these in the subsequent term clustering phase. This learning is based on the previous statistical approach.

The idea consists of combining two complementary iterative processes. On one hand, for a given iteration, we are looking for dependencies between governor/governed pairs that are considered most probable in the corpus. The probability of a dependency occurrence is labeled as `P(word1:c1,[label],word2:c2)`, being `word1` the governor word and `word2` the governed one, `c1` the lexical category of `word1` and `c2` that of `word2`, and `label`

the tag of the dependency. In this sense, each probability of lexical categories is considered. On the other hand, the second process computes, from the former one, the most probable semantic class to be assigned to the terms involved.

Both are computed in a similar way as the example explained in Table 1. We need to know a local probability and a global one. In each iteration, we reintroduce these values in the local one and look for both semantic and syntactic disambiguation, each profiting from the other. At the end, a fixed point assures the convergence of the strategy [7].

5 Conclusion and Future Work

The work described combines a number of NLP techniques in order to model concepts and relations between concepts contained in a text. Our aim is experimental and the goal is to introduce an architecture to generate a knowledge structure in order to develop question-answering facilities on textual documents.

In relation to previous proposals, we choose to work with a maximum degree of unsupervised tasks, which forces us to consider improved strategies in order to exactly identify both recurrent syntactic and stylistic patterns from text. The goal is to establish the linguistic context the parser will work with, in order to serve as a guideline for the later knowledge acquisition process.

References

1. Ferret, O.: Using collocations for topic segmentation and link detection. In: Proc. of the 19th Int. Conf. on Computational Linguistics, USA, vol. 1, pp. 1–7 (2002)
2. Habert, B., Naulleau, E., Nazarenko, A.: Symbolic word clustering for medium-size corpora. In: COLING, pp. 490–495 (1996)
3. Harris, Z.S.: Mathematical Structures of Languages. J. Wiley & Sons, USA (1968)
4. Jacquemin, C., Bourigault, D.: Term extraction and automatic indexing. Handbook of Computational Linguistics, 599–615 (1999)
5. Joshi, A.K.: An introduction to TAG. In: Mathematics of Language, pp. 87–114
6. Rousse, G., de La Clergerie, É.V.: Analyse automatique de documents botaniques: le projet Biotim. In: Proc. of TIA'05, pp. 95–104 (2005)
7. Sagot, B., de La Clergerie, É.V.: Error mining in parsing results. In: Proc. of the 21st Int. Conf. on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 329–336, Austrila (2006)
8. de La Clergerie, É.V.: DyALog: a tabular logic programming based environment for NLP. In: Christiansen, H., Skadhauge, P.R., Villadsen, J. (eds.) Constraint Solving and Language Processing. LNCS (LNAI), vol. 3438, Springer, Heidelberg (2005)
9. de La Clergerie, É.V.: From metagrammars to factorized TAG/TIG parsers. In: Proc. of IWPT'05, Canada, pp. 190–191 (2005)

XML Rules for Enclitic Segmentation*

Fco. Mario Barcala, Miguel A. Molinero, and Eva Domínguez

Centro Ramón Piñeiro, Ctra. Santiago-Noia km. 3, A Barcia,
15896 Santiago de Compostela, Spain
{fbarcala,molinero,edomin}@cirp.es

1 Introduction

Sentence word segmentation is an important task in robust part-of-speech (POS) tagging systems. In some cases this is relatively simple, since each textual word (or token) corresponds to one linguistic component. However, there are many others where segmentation can be very hard, such as those of contractions, verbal forms with enclitic pronouns, etc., where the same token contains information about two or more linguistic components. There are two main approaches to solving these difficult cases:

1. To treat tokens as a whole, and to extend the tagset to represent these phenomena. For instance, the Galician word `colléullelo` (*he or she caught it from them*) could be tagged as `Vei3s+Rad3ap+Raa3ms`, that is, verb (`V`), past simple (`ei`), 3rd person (`3`) singular (`s`) with an atonic pronoun (`Ra`), dative (`d`), 3rd person (`3`) masculine or feminine (`a`), plural (`p`) and another atonic pronoun (`Ra`), accusative (`a`), 3rd person (`3`), masculine (`m`), singular (`s`). Another widely alternative would be to tag `colléullelo` as `Vei3s2`, where the last `2` indicates that the verb has two enclitic pronouns.
2. To segment compound tokens, separating the components. For example, the same word can be broken into three parts: `colléu`, `lle` and `lo`. In this case, the components could be tagged separately as `Vei3s`, `Rad3ap` and `Raa3ms` respectively.

Although EAGLES [1] points towards using the second approach when these phenomena occur frequently, most papers and systems are based on the first approach [2] [3]. It is the simplest one, because there is no need to change current taggers [4], and it performs well with languages that hardly present these cases, for example English or French, but it presents several problems with others which have many occurrences of several linguistic components within the same word.

Such is the case of Galician, and to a lesser extent, other romance languages such as Spanish. In these cases, using the first approach, the tagset size would be greatly increased, and sometimes the POS tagging tasks would be impractical

* This paper results from a project developed at Centro Ramón Piñeiro para a Investigación en Humanidades and is partially supported by Ministerio de Educación y Ciencia (TIN2004-07246-C03-01), Xunta de Galicia (PGIDIT05PXIC30501PN) and Universidade de Vigo.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 273–281, 2007.
© Springer-Verlag Berlin Heidelberg 2007

due to the need for an extremely large training corpus in order to obtain an acceptable tagger behaviour [5]. Therefore, works which choose the first option do not explain how to solve other POS related problems: how to assign lemmas to compound tokens, how to use the system inside a more complex one (for instance, a translation machine or a parser), etc. which in other works concerning the second alternative are trivial processes.

In this paper we follow guidelines indicated by EAGLES, and we explain the internals of a highly configurable system which segments compound tokens into its components. It must be used in combination with a tagger which solves segmentation ambiguities [6] [7] so as to perform jointly all POS tagging related tasks.

We will center our attention on the hardest item in segmentation, that is, the processing of verbal forms with enclitic pronouns. Contrary to other ad-hoc systems, to do this we use easily configurable high level XML [8] rules outside the system code, which will allow linguists to adapt the system to their particular needs and languages. We have applied it to the Galician language [9], but the system is generic enough to be applied to most romance languages and any tagset.

2 Main System

Our system has to decide if a word is a verb with enclitic pronouns, and, if so, has to segment its different linguistic components and pretag them (with all their valid tags) to help the tagger to decide which tags are the correct ones. To do so, it needs:

1. A lexicon containing the greatest possible number of verbal forms with their tags and lemmas.
2. A lexicon containing the greatest possible number of verbal stems capable of presenting enclitic pronouns. It must include the stem, the different tags which it can take, and the lemma of each entry.
3. A lexicon with all the valid combinations of enclitic pronouns (the character sequences only).
4. A lexicon with all enclitic pronouns with their possible tags and lemmas.
5. A set of morphosyntactic rules which guide the segmentation.

It takes a long time to build these lexicons, particularly the first two, but their building process is simple and there are techniques to use them without computational problems [10].

The main system is responsible for performing the segmentation and calls the following two subsystems to do all its tasks:

– The verbal subsystem, which determines if segmentations proposed by the main system are correct. If so, it assigns tags to the verbal part, filtering out impossible ones.

– The enclitic pronoun subsystem, which assigns tags to the enclitic pronouns and filters out inappropriate ones.

In a more detailed way, first of all, the main system finds candidate words for a verb with enclitic pronouns. It processes the text word by word and calls the verbal subsystem only when it finds a word in which:

– The final or enclitic pronoun part is a valid combination of one or more enclitic pronouns, that is, it is in the third lexicon listed above.
– The initial or verbal part is in the lexicon of verbal stems which can present enclitic pronouns.

It then passes the possible segmentation to the verbal subsystem: the verbal part, and the enclitic pronoun part. Note that it could call the verbal subsystem several times for the same word, since there could be different possible segmentations. After this, it uses the enclitic pronouns lexicon to segment the enclitics and to call the enclitic pronouns subsystem for each one.

3 Verbal Subsystem

The verbal subsystem receives the verb part and the enclitic pronoun part from the main system and decides, by checking against its rules, if it is a valid segmentation. If so, it assigns candidate POS tags to the different linguistic components.

To avoid building an ad-hoc system, we externalize these advanced if-then rules through an XML file.

3.1 Verbal Rules Description

In figure 1 we show the XML DTD of this file. As can be seen, a rules document is a set of one or more rules, and each rule has one or more conditions (if) with their action (then).

A condition part has the following elements:

– **target**: the target of the evaluation of the rule. It can be *verb_part*, that is, the condition will be applied to the verbal part, *enclitic_part*, the condition will be applied to the enclitic pronoun part, *complete_form*, the condition will be applied to the complete form (the verb part and the enclitic pronoun part joined together) and *verb_tag*, the condition will be applied to the set of tags of the verbal part.
– **content**: the condition itself. It contains one or several *evaluation* elements with the evaluation expressions. The *evaluation* element has the *at* attribute to determine which portion of the target has to be matched with the expression. Several evaluation elements are interpreted as a logical OR between them.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT document (rule+,default_rule?)>
<!ELEMENT rule (condition+)>
<!ELEMENT default_rule (condition+)>
<!ELEMENT condition (target,content,action,check_default,filter*)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT content (evaluation+)>
<!ELEMENT action (#PCDATA)>
<!ELEMENT check_default (#PCDATA)>
<!ELEMENT filter (#PCDATA)>
<!ELEMENT evaluation (#PCDATA)>
<!ATTLIST evaluation at (all|end|begin) #REQUIRED>
```

Fig. 1. Verb rules DTD

- **action**: the action to be executed if the evaluation condition is true. It can be *continue*, that is, to continue evaluating the next condition, *accept*, which finishes the evaluation confirming that it is indeed a correct segmentation, *reject*, which finishes the evaluation confirming that it is not a correct transformation, and *filter*, which is the same as *continue* but removes the tags specified in *filter* elements.
- **check_default**: Determines if *default_rule* must be evaluated.
- **filter**: It specifies a set of tags that must be removed before continuing if the *action* element contains *filter*. It contains a boolean expression that is checked against all the tags of *verb_part*. Tags which match the boolean expression are removed and the system continues with its comparisons. This is useful when there are non-valid tags for current segmentation.

The subsystem tests the rules from top to bottom. If an *evaluation* expression of a *condition* matches the specified portion of the *target*, the corresponding *action* is executed. The process stops when there is an *accept* or a *reject* in an executed action or when all rules have been tested. If there is not a matching rule which rejects the segmentation, the default behaviour is to accept it.

4 Enclitic Pronoun Subsystem

The enclitic pronoun subsystem is called by the main system once for each enclitic pronoun. It receives the *verb_part*, the *enclitic_part* and the enclitic to be processed, and assigns possible POS tags for the enclitic being processed. It also works with XML rules similar to the ones above.

4.1 Enclitic Pronoun Rules

Pronoun rules could be developed in the same way as verb ones, but we have decided to add some modifications in order to make the design of these rules

easier. As we can see in the DTD in figure 2, the main differences with respect to verbal rules are:

1. There is no *default_rule*.
2. Rules have a *type*, which can be *last_enclitic*, that is, the rule is applied if the received enclitic is the last one of the *enclitic_part*, or *intermediate_enclitic*, where the rule is applied if the received enclitic is not the last one.
3. Inside each rule there is an *enclitic* element which contains the enclitic to which this rule will be applied.
4. The *evaluation* element has three additional possibilities: *first_enclitic*, *next_enclitic* and *last_enclitic*, to specify matchings towards the first or the last enclitic of the *enclitic_part* or to the *next_enclitic* with respect to which it is being evaluated.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!ELEMENT document (rule+)>
<!ELEMENT rule (enclitic, condition+)>
<!ATTLIST rule type (intermediate_enclitic|last_enclitic) #REQUIRED>
<!ELEMENT condition (target,content,action,filter*)>
<!ELEMENT target (#PCDATA)>
<!ELEMENT content (evaluation+)>
<!ELEMENT action (#PCDATA)>
<!ELEMENT filter (#PCDATA)>
<!ELEMENT evaluation (#PCDATA)>
<!ATTLIST evaluation at (all|end|begin|first_enclitic|
                         next_enclitic|last_enclitic) #REQUIRED>
```

Fig. 2. Enclitic pronoun rules DTD

So, an *action* is executed if the enclitic to be processed matches the *enclitic* element of a rule, its position satisfies *type* attribute, and *evaluation* elements are verified.

5 Complete Example

Let's suppose that the verbal rules file includes only the rule shown in figure 3, and the enclitic pronoun rules file includes only that in figure 4. Let's suppose also that `colléullelo` appears in the processing text. The main system analyzes it and concludes that it can be segmented into `colléu`, the *verb_part* and `llelo`, the *enclitic_part*. It reaches this conclusion because `colléu` is in the verb stems lexicon, and `llelo` is in the valid enclitic combinations lexicon.

The main system calls the verb subsystem, and it applies its rules. The rule in figure 3 says that if the verb part ends with `ei`, `éi`, `eu`, `éu`, etc., and the enclitic pronoun part starts with `o`, `lo`, etc., then it is not a valid segmentation.

```
<rule>
  <condition>
    <target>verb_part</target>
    <content>
      <evaluation at="end">
        ei OR éi OR eu OR éu OR ou OR óu OR iu OR íu OR ai OR ái
      </evaluation>
    </content>
    <action>continue</action>
    <check_default>no</check_default>
  </condition>
  <condition>
    <target>enclitic_form</target>
    <content>
      <evaluation at="begin">
        o OR lo OR a OR la OR los OR las OR -lo OR -la OR -los OR -las
      </evaluation>
    </content>
    <action>reject</action>
    <check_default>no</check_default>
  </condition>
</rule>
```

Fig. 3. Verbs rule example

In our case, `colléu` ends with `éu`, but the enclitic pronoun part does not start with the combinations contained in the second condition evaluation, so `colléu` as *verb_part* and `llelo` as *enclitic_part* is a valid segmentation (because there is no rule which says otherwise)[1]. As there is no *filter* element, the verb subsystem assigns all tags present in the verb stems lexicon (in this case only `Vei3s0` tag).

Finally, the main system calls the enclitic pronoun with each enclitic, first with `lle` and then with `lo`. The rule in figure 4 says that if `lle` appears in the middle of the enclitic part, if the following enclitic is `lo`, `la`, etc. then, tag `Rad3as` must be filtered (removed). That is, although in the enclitic pronouns lexicon `lle` appears with the two possible tags (singular, `Rad3as` and plural, `Rad3ap`), `llelo` is always *it from them*, and never *it from him or her*, so the singular tag must be removed.

So, the system concludes that `colléullelo` must be segmented into `colléu`, as `Vei3s0`, `lle`, as `Rad3ap` and `lo`, as `Raa3ms`

6 Additional Issues

For the purpose of clarity we have only described the basic functionality of the system, but it has many other possibilities:

[1] `Colléulo`, `colléua`, etc. would not generate valid segmentation alternatives, even though `lo` and `a` are valid enclitic combinations.

```
<rule type="intermediate_enclitic">
  <enclitic>lle</enclitic>
  <condition>
    <target>enclitic_part</target>
    <content>
      <evaluation at="next_enclitic">
        lo OR la OR los OR las OR -lo OR -la OR -los OR -las
      </evaluation>
    </content>
    <action>filter</action>
    <filter>Rad3as</filter>
  </condition>
  <condition>
    <target>enclitic_form</target>
    <content>
      <evaluation at="next_enclitic">
        NOT lo AND NOT la AND NOT los AND NOT las AND NOT -lo
        AND NOT -la AND NOT -los AND NOT -las
      </evaluation>
    </content>
    <action>filter</action>
    <filter>Rad3ap</filter>
  </condition>
</rule>
```

Fig. 4. Enclitic pronouns rule example

Grouping Conditions

There is a *resolution* element which can appear after the first condition to group other conditions. It allows the grouping of several conditions affected by the first one, minimizing the number of rules that have to be created.

True Rules

We can define rules without *evaluation* elements. These rules have no *target* or *content* and are always evaluated as true, their *action* element running in all cases, in the case of verbal rules, or if the enclitic to be processed satisfies *type* and *enclitic* elements, in the case of enclitic pronoun rules.

Wildcards

Some rule elements allow the use of the ? wildcard, which matches any character. This is the case of *filter* and *evaluation* elements.

External Functions

filter element has an optional parameter, *function*. If present, it allows external functions to be specified to perform different kinds of transformation. The

parameters of the functions are specified in a *param* attribute of the *filter* element. This issue is very helpful when carrying out specific treatments.

Word Reconstruction

So far, the system can split a verbal form with enclitic pronouns into the corresponding parts and assign possible tags to them. The final problem to solve is that the result chunks are not linguistically consistent. For instance, *colléu* is not a valid word when it appears alone, it must be *colleu*, without accent, tag *Rad3ap* refers to the isolated word `lles` and not to `lle`, and `lo` is an alomorph of `o`. So, the main system has to undo morphological changes which take place when the verbal part and the enclitic part are joined.

To do this for the enclitic pronouns, we use the external built-in function *replace* in the *filter* element of enclitic rules, which makes the replacement. *filter* rule element of figure 4 must be `<filter function=‘‘yes’’ param=‘‘lles’’>replace</filter>`, and the case of `lo` is solved by adding a simple true rule since in Galician `lo` is always, and only, an alomorph of `o`.

For the verbal part, the system makes use of lexicon lemmas. Once the verbal form has been segmented as we explained earlier, the system reconstructs the original form of the verb part using tags and lemmas of the resulting item to access these lexicons and to obtain the original form.

For instance, starting with the system result of the example in the previous section, first of all it obtains the lemma of `colléu`, looking into the verbal stems lexicon. Then, it searches tag `Vei3s0` and lemma `coller` in the verbal forms lexicon, obtaining the `colleu` form (without accent).

So, the final output of the system is `colleu`, as `Vei3s0`, `lles`, as `Rad3ap` and `o`, as `Raa3ms`

7 Conclusions

Sentence word segmentation is a very complex and important task in almost all natural language processing applications. Several works conceal or obviate the difficulties evolved in this process. In some cases, they adopt an easy partial solution acceptable for certain languages and applications, and, in others, they rely on a later or previous phase for solving it. However, there are hardly any papers with explanations describing how this later or previous phases have to be done.

In this paper we have described these problems, focusing on part-of-speech tagging tasks, and propose a solution for one of them: the segmentation of verbal forms which contain enclitic pronouns. We have presented a generic verb processing system, which segments and pretags verbs which have enclitic pronouns joined to them.

As we have seen, the system does not limit its function to segmentation, since it pretags the different linguistic components of a verbal form with enclitics, and removes invalid tags for its context. This innovative issue will be useful for

part-of-speech taggers, which can use this information to avoid making certain errors, thus improving its results.

Although we have applied it to the Galician language, it can be easily adapted to other romance languages. The generic rule system we have designed allows rules to be written on the basis of XML files. This, combined with the use of lexicons, makes this adaptation simple and independent of the system internals.

References

1. Expert Advisory Group on Language Engineering Standards (EAGLES). Synopsis and Comparison of Morphosyntactic Phenomena Encoded in Lexicons and Corpora. A Common Proposal and Applications to European Languages. In: EAGLES Document EAG-CLWG-MORPHSYN/R (May 1996)
2. Moreno, J.L.A., Lugrís, A.Á., Guinovart, X.G.: Aplicación do etiquetario morfosintáctico do SLI ó corpus de traducción TECTRA. Viceversa , 207–231 (2002)
3. Carreras, X., Chao, I., Padró, L., Padró, M.: FreeLing: An Open-Source Suite of Language Analyzers. In: Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC'04), Lisbon, Portugal (2004)
4. Brants, T.: A statistical part-of-speech tagger. In: Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP'2000), Seatle (2000)
5. Graña, J.: Técnicas de Análisis Sintáctico Robusto para la Etiquetación del Lenguaje Natural. In: Doctoral thesis, Universidad de La Coruña, Spain (2000)
6. Graña, J., Alonso, M.A., Vilares, M.: A Common Solution for Tokenization and Part-of-Speech Tagging: One-Pass Viterbi Algorithm vs. Iterative Approaches. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2002. LNCS (LNAI), vol. 2448, pp. 3–10. Springer, Heidelberg (2002)
7. Graña, J., Barcala, F. M., Vilares, J.: Formal Methods of Tokenization for Part-of-Speech Tagging. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 240–249. Springer, Heidelberg (2002)
8. World Wide Web Consortium, `http://www.w3c.org`
9. Álvarez, R., Xove, X.: Gramática da Lingua Galega. In: Editorial Galaxia, Vigo, Spain (2002)
10. Graña, J., Barcala, F. M., Alonso, M.A.: Compilation Methods of Minimal Acyclic Finite-State Automata for Large Dictionaries. In: Watson, B.W., Wood, D. (eds.) CIAA 2001. LNCS, vol. 2494, pp. 135–148. Springer, Heidelberg (2003)

Certified Genetic Algorithms: Crossover Operators for Permutations

F. Aguado, J.L. Doncel, J.M. Molinelli,
G. Pérez, C. Vidal, and A. Vieites

Departament of Computer Science
University of A Coruña
A Coruña, Spain
{aguado,doncel,molineli,gperez,eicovima,avieites}@udc.es

Abstract. In the context of Genetic Algorithms, the use of permutations for representing the chromosomes, instead of the most common binary encoding, has turned out to be more natural and convenient in order to resolve some optimization problems. With the purpose of adapting the classical crossover to this special representation, several proposals can be found in the literature [2,3,8]. In this paper we use Coq to formally implement some of these crossover operators and also to verify that they satisfy the required specifications. As we have considered permutations of (possibly) repeated elements, we can cover a wider collection of applications.

Keywords: Genetic Algorithm; Crossover; Coq; Calculus of Inductive Constructions; Specification; Theorem Proving.

1 Introduction

In this research we have combined Evolutionary Computation and Theorem Proving which are two different domains of Theoretical Computer Science. Specifically we have worked here with Genetic Algorithms (GAs) and the Proof Assistant Coq.

Genetic Algorithms were designed to mimic some of the processes observed in natural evolution in order to resolve optimization problems. They follow the general principle of biological evolution which says that better adapted individuals have more chances of surviving in future generations. The work of John Holland [4] is considered the origin of Genetic Algorithms theory whose foundations can be found in [6]. Although there are different types of these algorithms, some common processes are present in all of them: codification, evaluation and reproduction.

Usually they do not work directly with the original individuals of the population (phenotypes) using, instead, a new population of chromosomes (genotypes). The most common encoding technique is the use of strings $p = p_1 p_2 \dots p_n$ of binary bits ($p_i \in \{0, 1\}$).

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 282–289, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In each generation, the algorithm uses a fitness function to measure the worth of any individual, then multiple individuals are stochastically selected from the current population (based on their fitness), and they are modified with crossover and eventually mutation to obtain a new population. This new population is then used in the next iteration of the algorithm.

Mutation may cause changes in the biological component of the individuals (in a binary string mutation changes 0 to 1 and viceversa). On the other hand, crossover creates offspring by combining genetic material from the chromosomes of both parents.

In this paper, we concentrate on the crossover operator. The simplest version is the single point one. A crossover point on the parents is randomly selected and then heads and tails are exchanged.

This single-point crossover can be extended to the case of several cutting points. In [1], we implement in Coq a generalization of the multipoint crossover operator introduced by DeJong in [5].

Although binary representation is the most common encoding technique in the GA community, it is also true that other representations are often more natural and convenient to use. That is the case of the Traveling Salesman Problem (TSP) and the graph colouring problem (GCP) (further details can be found in [3]). In these kinds of applications the use of permutations of a finite list of elements have turned out to be a good option. But, in this case, the classical crossover can produce illegal offspring

In [2,8] several crossover operators specifically designed to adapt to this special encoding representation are introduced. Using them as an starting point, we have studied here the Order crossover (OX-crossover), the Position Based crossover (PBX-crossover) and the Order Based crossover (OBX-crossover).

In our research, we have used Coq in order to implement, in a functional language, some crossover operators, certify the correctness of this implementation and also prove some important properties satisfied. Thanks to this process, some close relations between these crossover functions have shown up. It is important to remark that we have considered here permutations of (possibly) repeated elements which is not the case in the common literature.

2 Preliminaries

Let D be a finite non-unitary nonempty set. We use lists of elements of D as chromosomes (in Coq, values of type `list D`), $p = p_1 \dots p_m$.

As the universes we are talking about are always sets of permutations, we need to have this concept formalized. For this purpose, we use the predefined predicate `permutation`. For convenience, we will denote $p \approx q$ for `permutation p q`.

We then implement the function `alelo_remove` in a recursive[1] way. This function *eliminates the first occurrence of an element a from a list p* (*if a appears in p*). For simplicity of notation, we write $p - a$ instead of `alelo_remove a p`.

[1] We use the keyword `Fixpoint` in the defintion of any recursive function.

```
Fixpoint alelo_remove (a: D) (p: (list D)) {struct p}:  (list D) :=
    match p with
    | nil => nil
    | a1 :: t =>   match ale_dec a a1 with
                   | left _ => t
                   | right _ => a1 :: alelo_remove a t
                   end
    end.
```

We define the predicate `incl_rep` which determines if, given two lists p and q of (possibly) repeated elements, p is included in q, which means that each element of p appears in q at least as many times as in p. For abbreviation, let $p \subseteq q$ stand for `inc_rep p q`.

```
Fixpoint incl_rep (p q: (list D)) {struct p}: Prop := match p with
     | nil => True
     | h :: t => In h q /\ incl_rep t (alelo_remove h q)
     end.
```

The following result expresses an important equivalence between the predicates `incl_rep` and `permutation` and sets up a new characterization for permutations.

Theorem 1. *Let $p, q \in$ list D. It holds $p \approx q$ if and only if $p \subseteq q$ and $q \subseteq p$.*

With the function `alelo_remove`, we implement `list_diff`; this is a Coq function that, given two lists p and q, *eliminates q from p*. Thus `list_diff p q` is the list obtained by deleting in `p` (by using `alelo_remove`) each element of `q`. In order to simplify notation, we will use $p \backslash q$ instead of `list_diff p q`.

Below, we introduce a result which relates the predicates `permutation` and `incl_rep` to the function `list_diff`.

Theorem 2. *If $p, q \in$ list D, then:*

1. *If $q \subseteq p$ and if $p \backslash q = [\,]$, then $p \approx q$,*
2. *If $p \approx q$, then $p \backslash q = [\,]$.*

The following theorem will be fundamental in the proofs of crossover properties.

Theorem 3. *Let $p, q \in$ list D satisfying $p \subseteq q$. Then $(q \backslash p \; +\!+ \; p) \approx q$.*

3 Order Crossover

In this section we implement in Coq a variant of the *Order crossover* designed by Davis [2].

Let us begin with an example[2] which illustrates how this crossover works. Let P_1 and P_2 ($P_1 \approx P_2$) be two chromosomes:

$$P_1 = a\ b\ c\ d\ e\ f\ g\ h\ i \qquad P_2 = h\ g\ d\ i\ a\ b\ e\ f\ c$$

[2] Like in the GA literature.

We choose randomly two natural numbers n and m (for instance $n = 4$ and $m = 7$). The substring $s(P_1)$, extracted from P_1, consists of the elements of P_1 that occupy the positions between n and m (in the example above, $s(P_1) = e\ f\ g$). We now delete in P_2 the alleles of $s(P_1)$, obtaining $P_2' = h\ d\ i\ a\ b\ c$. Finally, we insert in P_2' the substring $s(P_1)$ resulting in the permutation $H_1 = h\ d\ i\ a\ e\ f\ g\ b\ c$. Thus, after the crossover, we obtain a string which has between n and m the corresponding alleles from P_1 and, in the other positions, the alleles of the list P_2' which is obtained as the difference between the elements of P_2 and $s(P_1)$(keeping up their order).

Note that this crossover depends on P_1, P_2 and on two natural numbers n and m that determine the positions where $s(P_1)$ is extracted from P_1.

In order to implement in Coq this crossover, we begin with the definition of the substring $s_{n,m}(p)$ chosen in the list p by using the function `segm`. Here we need the functions $p \uparrow n$ (`first`) and $p \downarrow n$ (`cut`) already implemented in [1].

Definition 1. *Let $p = (p_1, \ldots, p_n) \in$ list D, where n is the length of p, and $m \in \mathbb{N}$. We define*

$$p \uparrow m = \begin{cases} (p_1, \ldots, p_m) & \text{if } m < n, \\ p & \text{if } m \geq n \end{cases} \qquad p \downarrow m = \begin{cases} (p_{m+1}, \ldots, p_n) & \text{if } m < n, \\ [\,] & \text{if } m \geq n \end{cases}$$

Definition 2. *Let $p \in$ list D and $n, m \in \mathbb{N}$. We define $s_{n,m}(p) = (p \uparrow m) \downarrow n$.*

With this function, the Order crossover of two lists p and q is defined as follows:

Definition 3. *Let $p, q \in$ list D and let $n, m \in \mathbb{N}$. We define*

$$\mathtt{ox}_{n,m}(p, q) = (q \setminus s_{n,m}(p)) \uparrow n \ +\!+ s_{n,m}(p) \ +\!+ (q \setminus s_{n,m}(p)) \downarrow n.$$

```
Definition ox_cross (n m: nat)(p q: (list D)) :=
 (first n (segm_diff n m q p))++(segm n m p)++(cut n (segm_diff n m q p)).
```

where `segm_diff n m p q = list_diff p (segm n m q)`.

The following result shows that the Order crossover satisfies the required specifications.

Theorem 4. *Let $p, q \in$ list D, where $p \approx q$, and let $n, m \in \mathbb{N}$ such that $n \leq m \leq$ length(p). Then, we have:*

1. $s_{n,m}(p) = s_{n,m}(\mathtt{ox}_{n,m}(p, q))$,
2. $q \setminus s_{n,m}(p) = (\mathtt{ox}_{n,m}(p, q)) \uparrow n \ +\!+ (\mathtt{ox}_{n,m}(p, q)) \downarrow m$.

Throughout this paper, since all chromosomes used for representing the possible solutions of a problem have to be permutations of a given list, we must show that this crossover is internal in any of these sets.

Theorem 5. *Let $n, m \in \mathbb{N}$ and let $p, q \in$ list D such that $p \approx q$. Then*

$$\mathtt{ox}_{n,m}(p, q) \approx p.$$

Proof. The proof of this theorem is obtained taking into account the result 3 in this paper, [1, theorem (1.13)] and also that, for any list r and any natural number n, it holds that $r = r \uparrow n \ +\!+ r \downarrow n$.

4 Position Based Crossover and Order Based Crossover

In this section we implement in Coq the *position based crossover* and the *order based crossover* operators due to Syswerda [8][3].

To understand how these operators work, let us see an example. Let P_1 and P_2 be two chromosomes ($P_1 \approx P_2$)

$$P_1 = a\ b\ c\ d\ e\ f\ g\ h\ i \qquad P_2 = h\ g\ d\ i\ a\ b\ e\ f\ c$$

The process to carry out the *position based crossover* is shown bellow:

1. Let us fix a binary list (possibly, chosen by chance) called from now on *crossover pattern* that marks some positions in the chromosomes. Let $\ell = 0\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0$ be the pattern. First of all, we extract from P_1 the substring corresponding to the positions where ℓ has ones. Thus, we get an intermediate schema:
$$H_1' = \square\ b\ \square\ d\ e\ \square\ \square\ h\ \square$$
2. We delete in P_2 the fixed alleles of H_1'. Hence we get the string $P_2' = g\ i\ a\ f\ c$.
3. To finish, we fill each hole in H_1' with the elements of P_2' respecting the order of it and resulting in the chromosome $H_1 = g\ b\ i\ d\ e\ a\ f\ h\ c$.

With respect to the *order based crossover*, we follow the same guidelines as the previous one except that the intermediate schema H_1' is built with the alleles of `ext` $\ell\ P_1$ occupying the corresponding positions[4] they have in P_2, but respecting the order of P_1; so $H_1' = b\ \square\ d\ \square\ \square\ e\ h\ \square\ \square$. The resulting chromosome is given by $H_1 = b\ g\ d\ i\ a\ e\ h\ f\ c$.

In order to formalize the crossover pattern, we define a set (`bit`) with two values representing 0 and 1. Each crossover pattern is an element of *list bit* type.

The fist step in the above process is implemented with the function `ext`. Given a string p and a crossover pattern ℓ, this function returns a string obtained by extracting from p the alleles corresponding to the positions where ℓ has ones.

```
Fixpoint ext (l: (list bit)) (p: (list D)) {struct l}: (list D):=
    match l with
    | nil => nil
    | a :: t => match a with
                |zero => match p with
                         |nil => nil
                         | _ :: lt => ext t lt end
                |one  => match p with
                         |nil => nil
                         | h :: lt => h :: ext t lt end
                end
    end.
```

[3] During this process of formalization, we have realized that the position based crossover and the so called uniform order crossover [3] are actually the same.

[4] Note that we can have several positions to choose because of the (possibly) repetition of the elements in a list.

Given two lists p and q and a crossover pattern ℓ, the function `merge` returns a new list obtained by substituting, in order, each zero of ℓ with an element of p and each one with an element of q; at the end, we complete the list (if necessary) with the rest of the elements of p.

```
Fixpoint merge (l: (list bit)) (p q: (list D)) {struct l}:(list D):=
   match l with
   | nil => p
   | a :: t => match a with
               |zero => match p with
                        |nil => merge t nil q
                        | h1 :: t1 => h1 :: merge t t1 q end
               |one  => match q with
                        |nil => merge t p nil
                        | h2 :: t2 => h2 :: merge t p t2 end
               end
   end.
```

In order to reach our objectives, we need the auxiliary functions `app` and `complement`. The first one computes the number of *zeros* or the number of *ones* in a pattern whereas the function `complement` returns the complementary pattern $\overline{\ell}$ of a given one ℓ.

An explicit and expected relation between the functions `ext` and `merge` is given by the following lemma.

Lemma 1. *Given any $p, q \in$ list D and any $\ell \in$ list bit suppose that length$(p) =$* `app` $0\ \ell$ *and that length$(q) =$* `app` $1\ \ell$*. Then:*

1. `ext` ℓ (`merge` $\ell\ p\ q) = q$,
2. `ext` $\overline{\ell}$ (`merge` $\ell\ p\ q) = p$.

Now we define the *position based crossover*.

Definition 4. *Let $p, q \in$ list D and let $\ell \in$ list bit. We define*[5]

$$\texttt{pbx}\ \ell\ p\ q = \texttt{merge}\ \ell\ (q \setminus \texttt{ext}\ \ell\ p)\ \texttt{ext}\ \ell\ p.$$

```
Definition pb_cross (l: (list bit)) (p q: (list D)) :=
   merge l (list_diff q (ext l p)) (ext l p)
```

For the implementation of the *order based crossover* and, since we want the alleles of `ext` $\ell\ P_1$ to remain in the same positions they were occupying in P_2, we first implement a function that, for any pair of lists S and T, provides a binary pattern with the same length of S and with ones in the positions of S corresponding to elements belonging to T.

[5] We have proved that the *position based crossover* generalizes the *order crossover*. It suffices to take a pattern ℓ with ones in all the positions between the parameters n and m of the *order crossover* and with zeros in the rest.

```
Fixpoint position (s t: (list D)) {struct s}: (list bit) := match s with
  | nil => nil
  | a :: r => match  (In_dec ale_dec a t) with
                | left _ => one :: (position r (alelo_remove a t))
                | right _ => zero :: (position r t) end
  end.
```

Now we can define the *order based crossover* as follows:

Definition 5. *Let $p, q \in$ list D and let $\ell \in$ list bit. We define*

$$\texttt{obx}\ \ell\ p\ q = \texttt{merge}\ (\texttt{position}\ q\ (\texttt{ext}\ \ell\ p))\ (q \setminus \texttt{ext}\ \ell\ p)\ (\texttt{ext}\ \ell\ p).$$

```
Definition ob_cross (l: (list bit)) (p q: (list D)) :=
  merge (position q (ext l p)) (list_diff q (ext l p)) (ext l p).
```

As we have done in the previous section, we have to check that these crossovers satisfy the required specifications. This means that the *position based crossover* has the alleles of $\texttt{ext}\ \ell\ p$ in the positions where the pattern ℓ has ones and that the *order based crossover* has the alleles of $\texttt{ext}\ \ell\ p$ in the positions occupied by these elements in q. The elements of $q \setminus \texttt{ext}\ \ell\ p$ occupy the rest of the positions (with the same order) of both crossovers. We can see that in the following theorem.

Theorem 6. *Let $p, q \in$ list D satisfying $p \approx q$, and let $\ell \in$ list bit such that $length(p) = length(\ell)$. Then, it can be proved:*

1. $\texttt{ext}\ \ell\ (\texttt{pbx}\ \ell\ p\ q) = \texttt{ext}\ \ell\ p.$
2. $\texttt{ext}\ \overline{\ell}\ (\texttt{pbx}\ \ell\ p\ q) = q \setminus \texttt{ext}\ \ell\ p.$
3. $\texttt{ext}\ (\texttt{position}\ q\ (\texttt{ext}\ \ell\ p))\ (\texttt{obx}\ \ell\ p\ q) = \texttt{ext}\ \ell\ p,$
4. $\texttt{ext}\ \left(\overline{\texttt{position}\ q\ (\texttt{ext}\ \ell\ p)}\right)\ (\texttt{obx}\ \ell\ p\ q) = q \setminus \texttt{ext}\ \ell\ p.$

The following result has turned out to be very useful in several proofs afterwards.

Theorem 7. *Let $\ell \in$ list bit and $s, t \in$ list D with $length(t) \leq \texttt{app}\ 1\ \ell$, then:*

1. $\texttt{merge}\ \ell\ s\ t \approx s\ +\!+\ t$
2. *If, additionally, $t \subseteq s$, then* $\texttt{merge}\ \ell\ (s \backslash t)\ t \approx s$

Proof. In order to prove 1., we now proceed by induction on ℓ.

i) If $\ell = [\,]$ then $length(t) = 0$ and we deduce that $\texttt{merge}\ [\,]\ s\ [\,] = s = s +\!+ [\,]$.
ii) If $\ell \neq [\,]$, take $\ell = h :: \ell_1$,
Take first $h = 0$. If $s = [\,]$, we obtain $\texttt{merge}\ \ell\ [\,]\ t = t = [\,]\ +\!+ t$.
If $s = a :: s_1$, then $\texttt{merge}\ \ell\ s\ t = \texttt{merge}\ (0 :: \ell_1)\ (a :: s_1)\ t = a :: \texttt{merge}\ \ell_1\ s_1\ t$.
Since $length(t) \leq \texttt{app}\ 1\ \ell = \texttt{app}\ 1\ \ell_1$, by induction hypotheses we can assert that $\texttt{merge}\ \ell_1\ s_1\ t \approx s_1 +\!+ t$.
Hence, $\texttt{merge}\ \ell\ s\ t \approx a :: (s_1 +\!+ t) = (a :: s_1)\ +\!+ t = s +\!+ t$.
In a similar way we can prove the other case $h = 1$.

We want that, after applying both crossovers to any pair of lists p and q with the property that $p \approx q$, the result should be a permutation of any of them.

Theorem 8. *Take $p, q \in$ list D, such that $p \approx q$, and let $\ell \in$ list bit, then:*

$$\texttt{pbx}\ \ell\ p\ q \approx p\ ,\ \texttt{obx}\ \ell\ p\ q \approx p.$$

Proof. Since $p \approx q$, we know that $\texttt{ext}\ \ell\ p \subseteq q$. Besides, we have that $length(\texttt{ext}\ \ell\ p) \leq \texttt{app}\ 1\ \ell$ and also that $length(\texttt{ext}\ \ell\ p) = \texttt{app}\ 1\ (\texttt{position}\ q\ (\texttt{ext}\ \ell\ p))$. Both facts are sufficient to deduce, from theorem 7, the desired results.

5 Conclusions

First of all, we have established a library (in Coq and Caml)[6] with basic operations on lists. Then, we have formalized the crossover operators and incorporated them into an ongoing genetic library. Afterwards, we have proved the required specifications for these functions and also some important relations among them. As we have considered permutations of (possibly) repeated elements, we have increased the range of applications.

Acknowledgments. This paper was partially supported by the Secretaría de Estado de Universidades e Investigación under research project TIN2006-15455-C03-02 and also by the Xunta de Galicia under REGACA 2006/38.

References

1. Aguado, F., Doncel, J.L., Molinelli, J.M., Pérez, G., Vidal, C.: Genetic Algorithms in Coq. Generalization and Formalization of the Crossover Operator, sumited to Theoretical Computer Science (2006)
2. Davis, L.: Applying Adaptive Algorithms to Epistatic Domains. In: IJCAI85, Proceedings of the Ninth International Joint Conference on Artificial Intelligence, pp. 162–164 (1985)
3. Davis, L.: Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York (1991)
4. Holland, J.H.: Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor (1975)
5. De Jong, A.K.: An Analysis of the Behavior of a class of Genetic Adaptive Systems, PhD Thesis, University of Michigan (1975)
6. Mitchell, M.: An introduction to Genetic Algoritms. MIT Press, Massachusetts (1996)
7. Paulin-Mhoring, C.: Extraction de programmes dans le calcul des constructions, Thése de doctorat, Université de Paris VII (1989)
8. Syswerda, G.: Schedule Optimization using Genetic Algorithms. In: [3], ch. 21, pp. 332–349
9. INRIA. The Coq proof assistant, `http://coq.inria.fr`

[6] Coq is written in the Objective Caml language, which is a variant of Caml and includes the possibility of extracting automatically [7] functional programs from the algorithmic content of the proofs.

Contextual Spelling Correction*

J. Otero[1], J. Graña[2], and M. Vilares[1]

[1] Department of Computer Science, University of Vigo
Campus As Lagoas s/n, 32004 Ourense, Spain
{vilares,jop}@uvigo.es
[2] Department of Computer Science, University of A Coruña
Campus de Elviña s/n, 15071 A Coruña, Spain
grana@udc.es

Abstract. Spelling correction is commonly a critical task for a variety of NLP tools. Some systems assist users by offering a set of possible corrections for a given misspelt word. An automatic spelling correction system would be able to choose only one or, at least, to rank them according to a certain criterion. We present a dynamic framework which allows us to combine spelling correction and Part-of-Speech tagging tasks in an efficient way. The result is a system capable of ranking the set of possible corrections taking the context of the erroneous words into account.

1 Introduction

A classic problem in spelling correction is how to rank the correction candidates provided by the spelling correction algorithm. Some systems avoid tackling this task by offering the user the possibility of choosing from a set of possible corrections for a given misspelt word. This is possible for interactive applications like word processors but, for most NLP tools, spelling correction is one of the first tasks to be performed and small and ranked sets of alternatives are welcome in order to minimize the overload in later phases.

In this work, we use two algorithms which implement two different strategies to explore a Finite State Automata, previously compiled from a lexicon, in order to recover typing errors by looking for alternative words closest to the misspelt one, that is, with the lowest edit distance [1]. The first of these was developed by Savary and performs an exhaustive exploration of the Finite State Automata finding all the words closest to the misspelt one. Alternatively, a regional look-up algorithm developed by Vilares *et al.* can be performed. This algorithm avoids the need for exhaustive look-up by using the topology of the FSA to limit the search space to the region where the error is detected. Given that the search

* Research supported by the Spanish Government under projects TIN2004-07246-C03-01, TIN2004-07246-C03-02 and the Autonomous Government of Galicia under projects PGIDIT03SIN30501PR, PGIDIT02SIN01E and "Rede Galega de Procesamento da Linguaxe e Recuperacin de Informacin".

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 290–296, 2007.
© Springer-Verlag Berlin Heidelberg 2007

space is limited the time and space cost decreases and the number of correction alternatives is usually lower.

In order to rank the repair alternatives offered by each of the two spelling correction algorithms some additional information is needed. A stochastic Part-Of-Speech tagger based on a Viterbi algorithm extension over second order Hidden Markov Models [2] allows us to take advantage of syntactic and lexical information embedded in probabilities of transition between tags and emission probabilities of words.

2 Spelling Correction Algorithms

Research in spelling correction has been progressively focussing on solving three problems, each one more difficult than the previous one [3]. The first one concerns the detection of words which do not belong to the language and nowadays is widely overcome. The second one concentrates on how to correct isolated word errors. In response to this, different techniques have been developed. The third problem lies in taking the context of words into account to detect, and also correct, misspelt words.

In relation with isolated-word error correction we will center our attention on dictionary-based techniches. From this point of view, the detection of an erroneous word becomes as simple as searching to see if a word is in a list or not, while the correction task can be seen as the search of all the words closest to a given one. The use of Finite State Automata to implement efficient scanners is a well-established technique [4]. The main reasons for compressing a very large dictionary of words into a Finite State Automata are that its representation of the set of words is compact, and looking up a word in the dictionary is very fast (proportional to the length of the word) [5]

2.1 Global Finite-State Error Repair Algorithm

A global finite-state error repair technique was introduced by A. Savary [6]. This technique was developed from a K. Oflazer's algorithm [7] which searches for all possible corrections of a misspelt word that are within a given edit distance threshold. To do this, when an error is detected in the input word, elementary edition operations [1] are admitted. It should be noted that there are several combinations of these edition operations which are equivalent (e.g. inserting a 'p' just before an 'r' and then deleting the 'r' is equivalent to replacing the 'r' by a 'p'). For this reason Savary's algorithm may reach the same correction candidate several times with the consequent increase in computational cost due to the repetitive exploration of some paths.

Savary's algorithm reduces the search space in the lexicon dynamically, retaining only the closest corrections and attempting to reach the first correction as soon as possible. However, this technique performs a global look-up because

[1] Insertion, deletion or replacement of a character or transposition of two contiguous characters.

the only restriction to avoid it is the edit distance threshold. This global look-up allows Savary's algorithm to guarantees that all the closest words for a given misspelling will be provided.

2.2 Regional Finite-State Error Repair Algorithm

Savary's algorithm, introduced in the previous subsection, guarantees that all the words in the dictionary with the minimum edit distance to an erroneous word are supplied. It can be seen as an advantageous property because the correct word has more chance to be in the set of given alternatives. Nevertheless global strategies penalize the time and space cost and a bigger set of alternatives does not allways facilitate the choice of the correct one. It should be taken into account that the larger the number of possibilities, the more unlikely it is that the correct one will be choosen.

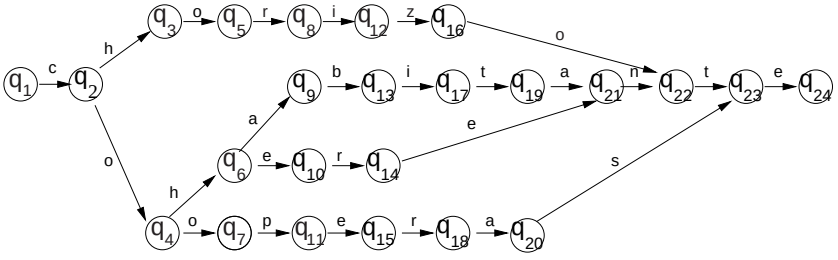


Fig. 1. Regions in a Finite State Automata

A regional finite-state error repair algorithm was suggested by Vilares *et al.* [8] to reduce both the computational cost and the number of correction alternatives associated to global strategies. The main difference with Savary's proposal is that the search space in the finite state automata is limited to the region where the error is detected. Let us use Fig. 1 to explain intuitively how the algorithm works. This figure represents a finite state automata which recognizes the following Spanish words: "*chorizo*" (sausage), "*cohabitante*" (a person who cohabits with another one), "*coherente*" (coherent) and "*cooperase*" (I cooperated). If the system tries to recognize the erroneous word "*cohenente*" it halts on an error in the state q_{10} because there is no transition for the first 'n'. At this point the search space is limited to the region which contains the state where the error is detected, in this case, the one that begins at state q_6 and ends at state q_{21}. In that situation, by replacing the first 'n' by an 'r' the system is able to recognize the word "*coherente*" and it is not necessary to look for more correction alternatives, thus avoiding a global look-up. Note that in this example a global look-up would not be very expensive but if we imagine a finite state automata to recognize a lexicon of thousands of words the computational saving would be important.

A very interesting property of this regional approach is that, in the worst case, it is equivalent to a global one. To illustrate it we use a new example. Let us consider the erroneous input string "*cohrizo*", resulting from transposing 'h' with 'o' in "*chorizo*". In this context, the system comes to a halt at state q_6 and no transition is possible on 'r'. So, the system looks for regional repairs from state q_4 to state q_{23} (the error region). In this case, there are no possible regional repairs which allow us to get to state q_{23}. As a result, the error region is extended to the one which begins at state q_2 and ends at state q_{23}, where the least cost regional repair is now defined as the transposition of the 'h' with 'o', to obtain the correct word "*chorizo*" (sausage), which agrees with the global repair because the error region fits with the entire automata.

The regional approach, unlike the global one, does not guarantee that all the closest words for a given misspelt will be provided because sometimes only the region of the finite state automata where the error has been detected will be explored.

3 Part of Speech Tagging

In the context of part-of-speech tagging, we center our attention on stochastic methods, particularly those based on Hidden Markov Models (HMMs), in which the classic version of the Viterbi algorithm [9] is applied on trellises (see Fig. 2), where the first row contains the words of the sentence to be tagged, and the possible tags appear in columns below the words. But trellises can not represent the segmentation ambiguity which is quite a frequent phenomenon in languages such as Spanish or Galician.

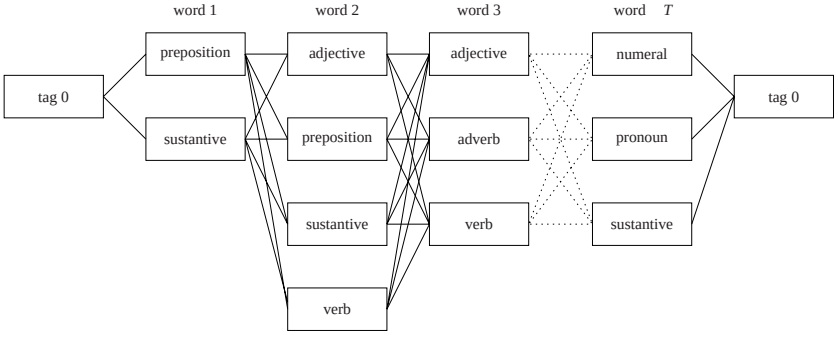


Fig. 2. A classical trellis

In [2], a tagger is developed which can not only assign tags to every token, but also decide whether some of them form or not the same term, according to different segmentation alternatives. For this task, a dynamic version of the Viterbi's algorithm is designed which can evaluate streams of tokens of different

lengths over the same structure taking advantage of the sharing of computations for common substreams. This structure takes the form of be a lattice, instead of a classical trellis.

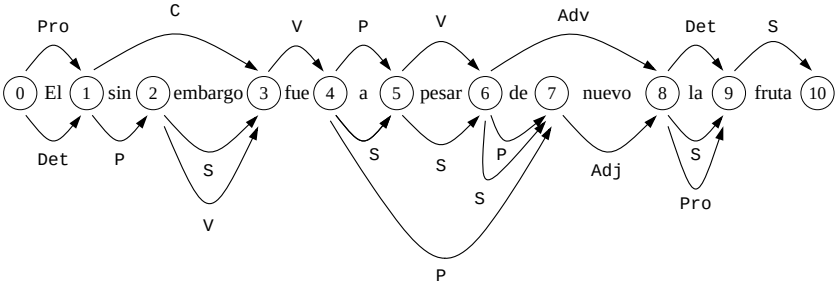


Fig. 3. A lattice structure

Fig. 3 shows a lattice for a singular Spanish sentence containing 20 arcs from which 234 possible paths can be built. In this case, the correct tagging is the one formed by the 8 arcs drawn in the upper part of the lattice, and its literal translation would be: *He however went to weights again the fruit.*

Given that segmentation alternatives can not be represented in trellises, in our running example 6 different trellises would be needed and the classic Viterbi algorithm would have to be applied over each one of them. Although the 6 trellises would represent different segmentation alternatives it should be noted that some computations would be repeated in all of them. Using lattices, the segmentation alternatives can be represented in the same structure and then, by mean of an adaptation of the Viterbi algorithm equations, the probability of each path can be computed. Finally, probabilities computed for different length paths must be normalized, according to their lengths, obtaining the best segmentation alternative with the corresponding POS tags.

4 Combining Spelling Correction and PoS Tagging

We can observe some similarities between segmentation disambiguation and contextual spelling correction. The first aims to choose the best segmentation alternative while the second aims to choose the best correction alternative and, in both cases, only contextual information extracted from adjacent tokens is taken into account. So lattices also constitute an excellent dynamic framework for the contextual spelling correction task.

In order to use lattices for contextual spelling correction, a slight modification must be made in relation to the lattice shown in Fig. 3, where arcs are only labeled with the POS tag corresponding to the words covered by it. In contextual spelling correction, arcs must be labelled with *word/tag* pairs because more than one correction alternative has to be represented in the same gap.

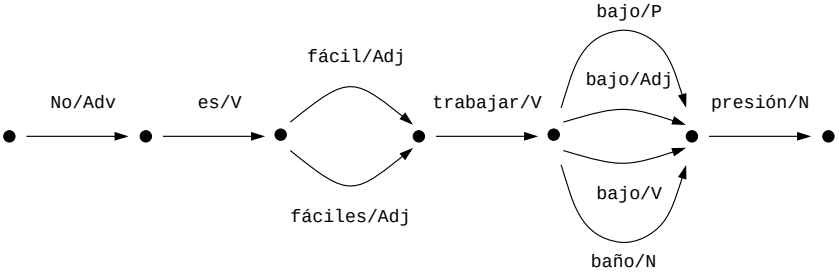


Fig. 4. Spelling correction alternatives represented on a lattice

Let us consider e.g. the sentence `No es fácile* trabajar baio* presión` (*It is not easy to work under pressure*), where the words `fácile` and `baio` are misspellings. Let us assume that our spelling corrector provides `fácil/Adj`, and `fáciles/Adj` as possible corrections for `fácile*`, and `bajo/Adj`, `bajo/P`, `bajo/V` and `baño/N` for `baio*`. Fig. 3 shows a lattice representing this situation. An execution of the dynamic Viterbi's algorithm over this lattice provides us with the tags of the words, and also the most probable spelling corrections in the context of this concrete sentence. Depending on the application, a ranked list of correction candidates may be required. To provide this, computed probability for each path in the lattice may be used.

5 Experiments

Our experiments have been performed on a set of sentences of the Spanish CLIC-TALP corpus [11], in which we have introduced random *human-like* errors by means of the tool MISPLEL [10].

In a first set of tests, we introduce 1 mistake per sentence. In this case, the average number of paths that have to be evaluated by the tagger is 5 per sentence, and the results for the two spelling correction algorithms are: the correct word is present in the list of alternatives provided by the global spelling corrector in 96.3% of cases (95% for regional approach); in those situations, the correct word is selected by the tagger in 93.9% of cases; therefore, by combining these indicators, we obtain a global performance of 90.5% of the sentences completely recovered (89.6% for regional approach).

In a second set of tests, we introduce at most 3 mistakes per sentence. In this case, the average number of paths to be evaluated by the tagger increases to 84 paths per sentence, and the percentage of the sentences that are completely recovered decreases to a still respectable 80.9% (78.6% for the regional approach).

In relation to the spelling correction algorithm, we observe some interesting data. The global strategy obtains a slightly better global performance but if we center our attention on those cases where both algorithms return the correct word among the set of alternatives, the percentage of correct choices by the tagger is greater for the regional approach because the set of alternatives provided is smaller.

6 Conclusion and Future Work

We have illustrated how spelling correction and part-of-speech tagging can be successfully combined, with the first task taking good advantage of this context-sensitive approach. However, the task of part-of-speech tagging is affected by an important overload of work, due to the large number of paths to be explored in every lattice when the different spelling correction alternatives are integrated. For this reason, our main future work will focus on the design of strategies to decide when it is possible to apply our least-cost regional error repair technique, keeping the performance at a similar level to the global one, but reducing the number of spelling correction alternatives. and therefore the overload of work in the tagger.

The combination between spelling correction and part-of-speech tagging carried out in this work seems to be particularly appropriate for dealing with spelling errors caused by the splitting or joining of words given that errors of this kind cause a different segmentation of the phrase where they appear.

References

1. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Doklady Akademii Nauk SSSR 163(4), 845–848 (1965)
2. Graña, J., Alonso, M., Vilares, M.: A common solution for tokenization and part-of-speech tagging. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2002. LNCS (LNAI), vol. 2448, pp. 3–10. Springer, Heidelberg (2002)
3. Kukich, K.: Technique for automatically correcting words in text. ACM Computing Surveys 24(4), 377–439 (1992)
4. Aho, A.V., Sethi, R., Ullman, J.D.: Compilers: principles, techniques and tools. Addison-Wesley, MA (1985)
5. Hopcroft, J.E., Ullman, J.D.: Introduction to automata theory, languages and computations. Addison-Wesley, Reading, MA (1979)
6. Savary, A.: Typographical nearest-neighbor search in a finite-state lexicon and its application to spelling correction. In: Watson, B.W., Wood, D. (eds.) CIAA 2001. LNCS, vol. 2494, pp. 251–260. Springer, Heidelberg (2003)
7. Oflazer, K.: Error-tolerant Finite State Recognition with Applications to Morphological Analysis and Spelling Correction. Computational Linguistics 22 (2006)
8. Vilares, M., Otero, J., Graña, J.: Regional finite-state error repair. In: Domaratzki, M., Okhotin, A., Salomaa, K., Yu, S. (eds.) CIAA 2004. LNCS, vol. 3317, pp. 269–280. Springer, Heidelberg (2005)
9. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. IEEE Trans. Information Theory IT-13 (1967)
10. Bigert, J., Ericson, L., Solis, A.: Missplel and Autoeval: two generic tools for automatic evaluation. In: Proceedings of Nodalida-2003, Reykjavik, Iceland (2003)
11. Sebastián, N., Martí, M.A., Carreiras, M.F., Cuetos, F.: LEXESP: léxico informatizado del español. Edicions de la Universitat de Barcelona (2000)

Multiple Label Text Categorization on a Hierarchical Thesaurus

Francisco J. Ribadas[1], Erica Lloves[2], and Victor M. Darriba[1]

[1] Departamento de Informática
University of Vigo
Campus de As Lagoas, s/n, 32004
Ourense, Spain
ribadas@uvigo.es, darriba@uvigo.es
[2] Telémaco, I. D. S., S.L.
Parque Tecnológico de Galicia
Ourense, Spain
ericalloves@gmail.com

Abstract. In this paper we describe our work on the automatic association of relevant topics, taken from a structured thesaurus, to documents written in natural languages. The approach we have followed models thesaurus topic assignment as a multiple label classification problem, where the whole set of possible classes is hierarchically organized.

1 Introduction

In many document processing tasks a correct identification of relevant topics offers a helpful starting point to develop advanced applications to deal with browsing and searching in large collections of documents. In this context, one of the most valuable tools are specialized thesauri. These kinds of structure organize a set of concepts relevant to a given domain in a hierarchical structure, making it possible to employ a sort of controlled vocabulary to simplify document processing.

The classical approach [2] relies on human processing to perform thesaurus term selection after reading the whole documents. This approach requires the availability of trained experts and suffers from a lack of scalability, since this kind of work is very time consuming and difficult to apply on large collections of documents. We propose to partially replace this kind of human made task with an automatic tool able to identify, for each input document, a list of potential descriptors taken from the domain thesaurus.

In this paper we describe our proposal on the automatic association of relevant topics, taken from a structured thesaurus, to documents written in natural languages. In our case we are interested in the domain of legislative texts in Spanish. We have an available thesaurus, manually built, with more than 1800 concepts, arranged in a tree structure. We also have a collection of legislative documents, whose main topics have been identified by humans according to the entries in that thesaurus.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 297–304, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The approach we have followed models thesaurus topic assignment as a multiple label classification problem, where the whole set of possible classes is hierarchically organized. Many previous proposals [11] have dealt with text categorization, but the case of hierarchical classes is usually omitted [8] or the generalization to multiple label classification is not directly supported [3,4].

Our aim is to build a system able to assign descriptive topics to input documents. The set of possible topics is taken from the thesaurus entries and for each document many descriptors may be selected with no special restrictions about the relationships among them. So, in the set of assigned descriptors we could find pairs of sibling entries or any combination of ancestors and descendants.

We also want that our system models in some way the processing made by humans when they perform this kind of tasks. In our domain, legal texts in Spanish, a very restricted kind of document structures uses to be commonly employed. Document contents can be segmented into consistent text regions and expert users pay special attention to those specific portions which usually carry the most relevant content. Examples of this kind of regions are the document introduction, the description of the document aims or the destination section. Also, human experts tend to use the thesaurus structure as a guide to select descriptive topics from it. So, we maintain this two intuitions in our approach that filters entries from the topics hierarchy in a top-down fashion.

The article is outlined as follows. Section 2 introduces our classification framework. Next, Section 3 describes the document representation and processing. In Section 4 the most relevant details about the training and classification strategies are described. Section 5 shows the results obtained in our preliminary experiments. Finally, Section 6 presents our conclusions and future work.

2 Classification Framework

Some questions need to be taken into account before starting to describe our proposal. First of all, we are working on a big domain form a text processing point of view. On one hand, we have a very large collection of legislative documents. These documents use to be quite long, with sizes ranging from hundreds to thousands of words. On the other hand, we also have a big set of potential classes arranged in a tree. In this context there are two main aspects to have in mind. First of all, we must ensure a practical computational cost, both in the training phase and specially in the topic assignment phase. We also must offer a robust classification framework, able to return a consistent list of topics for a great variety of input documents, without contradictions. With regard to the available resources, we have a thesaurus built by hand for the domain of legislative text and a set of historic documents with their corresponding descriptors assigned by human experts, which will be employed in the training phase.

With that premises in mind, a first approach could be to take the available documents and train a big classifier using all of the topics in the thesaurus as output classes. This approach is almost impractical from a computational cost point of view, but also it has many important problems with output quality and

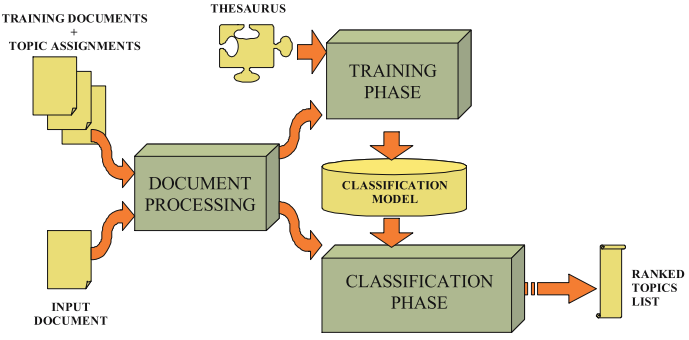


Fig. 1. Classification framework

a lack of robustness and consistence. Training such a classifier will suppose to estimate a large number of parameters with too many irrelevant features that will disturb the classification decision.

The strategy we have chosen is inspired by Koller and Sahami's work [7] and takes advantage of the class hierarchy to simplify the classification task in two aspects. Firstly, the global classification problem is reduced to a sequence of partial classifications, guided by the structure of our topic tree. Secondly, the computational cost for each classification step is reduced and the resulting quality is improved by means of the use of a specific set of features, exclusive to each node in the hierarchy. In this maner, the classification decision is distributed over a set of partial classifiers across the topics tree. In this model each internal node will be responsible for a local classification decision, where only a small set of features from the document will be taken into account to select the most promising descendants, which will repeat that process.

The main difference with Koller and Sahami's proposal is the final output of our classifier. Our aim is to get a set of relevant topics taken from the thesaurus, ordered according to their relevance, instead of a single class. We replace the greedy approach used in the original work, where only the best successor for each node was considered at each step. In our proposal, we proceed level by level, and all of the paths starting at successors with higher evaluation values are taken into account, and they are followed until no further expansion is admissible. The final set of topics is composed of those class nodes, ranking them according to the strength of the classification steps that lead to them.

In Fig. 1 we show the main phases in our approach, where three main components are outlined. Document processing, which is applied both in training and classification, has the responsibility of cleaning the documents to reduce the number of features employed to represent them. In the training phase, the training set of documents are taken and the topic hierarchy is traversed top-down, performing at each node local feature selection and training the corresponding partial classifier. Finally, in the classification phase, for each input document the topic tree is traversed top-down using the trained classifiers to decide whether

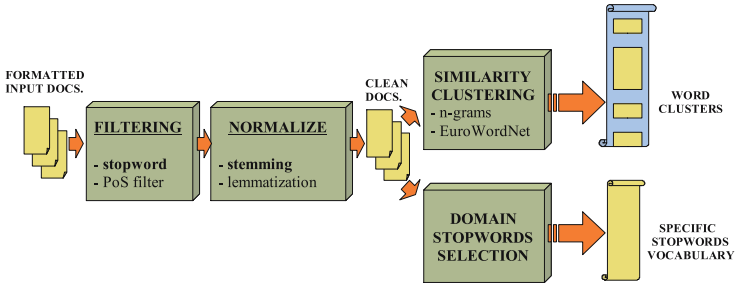


Fig. 2. Collection preprocessing

the corresponding topic is suitable to be taken as final descriptor and to make routing decisions to select one or more branches to continue searching. At the end, the list of potential descriptors for that document will be ranked and returned to the user.

3 Document Representation and Processing

Since this is a first approach to this kind of problem, we have tried to avoid using complex linguistic resources, like taggers [5], lemmatizers or shallow parsing [12]. Original documents were in HTML and PDF format and the first step was to extract plain text from them. Those text files were previously preprocessed to segment their text into regions and to identify which of those regions are relevant and suitable to extract descriptors from them.

A first processing, shown in Fig. 2, is performed on the whole collection. To omit non-relevant words we use a generic stop-word list for Spanish. Remaining words are normalized by means of stemming rules to overcome lexical variation problems. Once all of the the documents in the collection have been cleaned two structures are built. A specific stopword list, containing a vocabulary of commonly used words in the considered domain, that, in our case, makes possible to get rid of words frequently employed in legislative texts. A sort of dictionary of similar words, that allows us to identify groups of related words, is also built. We have employed a method to detect similar tokens at orthographic level by means of a hierarchical clustering algorithm which uses a n-gram based distance between word characters.

Both in training and classification, the list of features to be employed is extracted from cleaned documents in the way shown in Fig. 3. From the relevant regions of the input document domain specific stopwords are deleted. Optionally, some words that appears as labels in the thesaurus topics can be recovered to be taken into account as features. This features have demonstrated to be useful when short documents are processed. The surviving features will suffer a sort of semantical normalization using the similar word clusters built from the whole training collection. After that we obtain the list of features that will describe the input document in the training and classification phases.

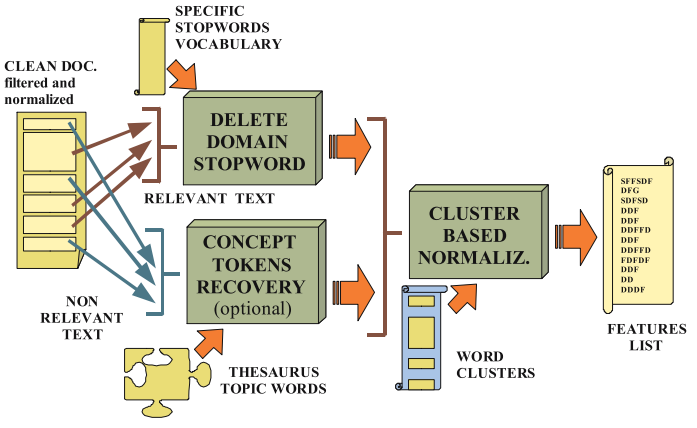


Fig. 3. Document processing and representation

4 Training and Classification

In this section we show the main components that conform our proposal. Once the set of training documents have been processed they are employed to train our hierarchical categorization model. This model contains for each thesaurus node the set of features with higher discrimination power at that level and a trained partial classifier to make the routing decisions. The idea behind this strategy is that using this set of classifiers decisions will be more precise and the overall classification quality will be improved.

4.1 Training Phase

In the training phase we take the whole training collection with the set of topics associated to each document and we traverse the topic tree, performing two tasks at every thesaurus entry. For each node, the subset of documents with at least one descriptor being a descendant of the current concept is selected. With these documents we apply simple feature selection techniques to find a set of features with the highest discrimination power among the different branches starting at this node. Then, a specialized classifier is trained to select the most promising branches at current level.

For each document a feature vector is built. Only the relevant stems selected for the current concept are employed, using their *tf-idf* [10] as feature values. The class for this training vector will be the current topic, if it is actually associated with the document, or the label of one of its sons, if some topic associated to the document falls into that branch. We have employed the WEKA machine learning engine [13] to train the specialized classifiers for each topic in our hierarchical thesaurus. We have tested several classification algorithms to be employed inside this hierarchical categorization scheme, as it can be seen in the experimental results section.

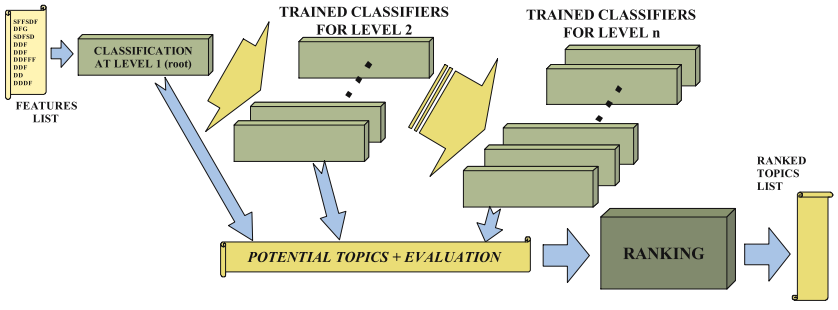


Fig. 4. Hierarchical classification

4.2 Classification Phase

Once all of the partial classifiers have been trained, the assignment of topics to new documents means traversing the thesaurus tree, as it is shown in Fig. 4. Starting at the thesaurus root, the feature vector for the document is built using the selected features for each node, and the most promising branches according to the partial classifier results are followed.

The original proposal by Koller&Sahami defines a single class output. They perform a greedy search selecting at each node only one class and stopping when a leave is reached. Since we are interested in multilabel classification, we have added two new components in our classification strategy. In this way, node classifiers have two missions. The first one is to detect if a topic is suitable to be considered as final descriptor, and the second one is to make a routing decision to determine the next steps in the search.

The routing decisions taken at each node are controlled by simple thresholds that take into account both the number of alternatives at each node and the strength of the potential classes returned by the classifier. If the class for the current topic has an evaluation value higher than this threshold it is considered to be suitable as a topic for describing this document. When a leave is reached or no successor classes have sufficient strength, the deeping is stopped. The final list of potential topics is ranked according to the set of values obtained in the sequence of partial classifications that lead to them. Different formulae, average, maximum or product, can be used to combine the strength values obtained in the path of classifications form the root to that descriptor.

5 Experimental Results

To illustrate our proposal we will review some preliminary experiments we have perform to test our method. In these experiments we have employed a portion of the legal corpus donated by Telemaco, S.L., with 2333 legislative documents with their corresponding set of descriptors assigned by human experts. These descriptors where taken from a set of 1873 thesaurus topics about the fields of agriculture, livestock and fishing. This corpus was randomly splitted to build a

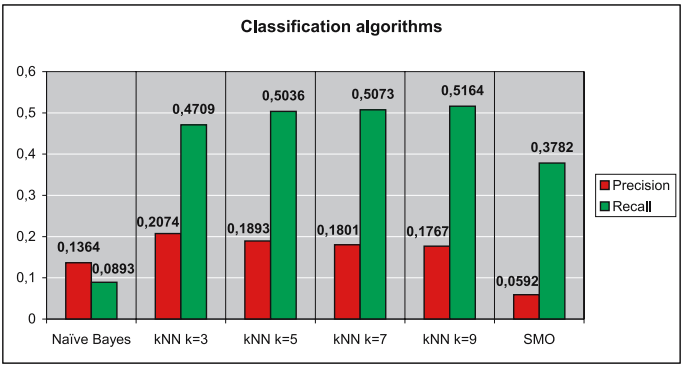


Fig. 5. Evaluation of classification algorithms

training data set with 2124 documents and a test dataset with 209 documents. To evaluate the experimental results we have employed two well known measures in the Information Retrieval field, precision and recall, using a modified version of the standard `trec_eval` [1] tool to compute them.

In Fig. 5 we show the average precision and recall values obtained in a set of experiments to test the using of different machine learning algorithms to perform the partial classifications across the thesaurus tree. We have tested a Naive-Bayes implementation [6], a k-Nearest Neighbors(k-NN) [1] learning method, with different values for k, and a Support Vector Machine model using Sequential Minimal Optimization(SMO) [9], all of them are included in the WEKA machine learning engine [13]. As it can be seen, the best results, both in precision and recall, where obtained with the k-NN method, using seven neighbors. In a deeper review of the descriptors obtained in that run, our approach used to offer better results when dealing with the most general topics, but it was unable to get a human level performace with the most specific descriptors.

6 Conclusions and Future Work

In this article we have proposed the use of a hierarchical multilabel classification approach which allows us to face the thesaurus topic assignment problem. We have followed a very flexible method, easy to be adapted to deal with different practical domains and to integrate several classification and text processing algorithms. The developed system offers quite good performance on average documents, even being able to avoid some human inconsistencies. When complex or very specific documents are processed, our tools are unable to work at human expert level, opening a field for further improvements.

With respect to future work, several aspects should be studied in our classification approach. Firstly, we intend to extend our experiments to other domains

[1] `http://trec.nist.gov/trec_eval`

and languages, in order to test its generality. Secondly, we aim to improve the system by integrating more powerful natural language processing tools.

Acknowledgements

The research reported in this paper has been partially supported by Telémaco, Información, Documentación y Sistemas, S.L. (`http://www.telemaco.com`), Xunta de Galicia(PGIDIT05SIN044E, PGIDIT05PXIC30501PN, *Galician Network for Language Processing and Information Retrieval*), Ministry of Education and Science (TIN2004-07246-C03-01) and University of Vigo

References

1. Aha, D., Kibler, D.: Instance-based learning algorithms. Machine Learning 6, 37–66 (1991)
2. Choi, J.H., Park, J.J., Yang, J.D., Lee, D.K.: An object-based approach to managing domain specific thesauri: semiautomatic thesaurus construction and query-based browsing. Technical Report TR 98/11, Dept. of Computer Science, Chonbuk National University (1998)
3. Chuang, W., Tiyyagura, A., Yang, J., Giuffrida, G.: A fast algorithm for hierarchical text classification. In: Kambayashi, Y., Mohania, M.K., Tjoa, A.M. (eds.) DaWaK 2000. LNCS, vol. 1874, pp. 409–418. Springer, Heidelberg (2000)
4. Dumais, S., Chen, H.: Hierarchical classification of Web content. In: Proc. of ACM-SIGIR-00, 23rd ACM Int. Conf. on Research and Development in Information Retrieval, pp. 256–263. ACM Press, New York (2000)
5. Grana, J., Alonso, M.A., Vilares, M.: A common solution for tokenization and part-of-speech tagging: One-pass Viterbi algorithm vs. iterative approaches. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2002. LNCS (LNAI), vol. 2448, Springer, Heidelberg (2002)
6. John, G.H., Langley, P.: Estimating Continuous Distributions in Bayesian Classifiers. In: Proc. of the Eleventh Conference on Uncertainty in Artificial Intelligence, pp. 338–345. Morgan Kaufmann, San Francisco (1995)
7. Koller, D., Sahami, M.: Hierarchically classifying documents using very few words. In: Proc. of 14th Int. Conf. on Machine Learning, Nashville, US, pp. 170–178 (1997)
8. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Learning to classify text from labeled and unlabeled documents. In: Proc. of the 15th National Conference on Artifical Intelligence, AAAI-98 (1998)
9. Platt, J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods - Support Vector Learning, MIT Press, Cambridge (1998)
10. Salton, G.: Automatic text processing. Addison-Wesley Longman Publishing Co., Inc., Boston, MA (1988)
11. Sebastiani, F.: Machine learning in automated text categorization. In: ACM Computing Surveys, vol. 24(1), pp. 1–47. ACM Press, New York (2002)
12. Vilares, J., Alonso, M.A.: A Grammatical Approach to the Extraction of Index Terms. In: Proc. of International Conference on Recent Advances in Natural Language Processing, pp. 500–504 (2003)
13. Witten, I., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)

A Formal Foundation for Knowledge Integration of Defficent Information in the Semantic Web*

Joaquín Borrego-Díaz and Antonia M. Chávez-González

Departamento de Ciencias de la Computación e Inteligencia Artificial.
E.T.S. Ingeniería Informática-Universidad de Sevilla.
Avda. Reina Mercedes s.n. 41012-Sevilla, Spain
{jborrego, tchavez}@us.es

Abstract. Maintenance of logical robustness in Information Integration represents a major challenge in the envisioned Semantic Web. In this framework, it is previsible unprecise information (with respect to an ontology) is retrieved from some resources. The sound integration of such information is crucial to achieve logical soundness. We present a *data-driven* approach to classify that knowledge by means of the *cognitive entropy* of the possible robust ontology extensions and data.

1 Introduction

Knowledge Integration is a major issue in both Knowledge and Data Engineering (KDE) and Artificial Intelligence fields. Therefore, it has to be solved in one of the current projects where both fields come together, the Semantic Web (SW). In this framework, there are many situations where defficent information obstructs the use of trustworthy reasoning systems [1]. Even, it can suggest the revision of the intensional component of the Knowledge Database, namely the ontology.

In Ontological Engineering, an accurate classification of the objects is a main goal. It considers that the individuals involved in such data will remain well classified when they fall in the most specific classes of the concept taxonomy. A solution for that classification may be to introduce *provisional concepts* or *notions* for classifying individuals. Since the insertion of a notion of this kind is mainly *data-driven*, the notion is initially located in lower levels of the taxonomy of concepts. This is like that because very little is known about its definition, as well as how to subclassify their elements. In any way, we need to build a *robust extension* of the ontology to trustworthy work with the new concepts.

The subject of this paper is to present a method to insert concepts which have been induced by defficent information, into an ontology. Since data that suggests the revision is unprecise (up to certain degree), the user is not interested in to obtain a definition of the new concept. That is, one only aims to provisionally classify facts waiting for more precise information. This *data-driven* approach is investigated here. The method proposed for ontological insertion lies in extending

* Supported by project TIN2004-03884 of Spanish Ministry of Education and Science, cofinanced by FEDER founds.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 305–312, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the ontology to provisionally classify the individuals. The extension preserves main features of ontology source, and it can be considered as a robust extension (*lattice categoricity* [2], [3]). The main benefit of the method lies in the fact of it is fully formalized and it is semi-automated, assisted by automated reasoning systems.

There is other use case requiring this kind of ontological insertion. When the ontology engineer identifies a set of specific data (that is, facts on most specific concepts of the ontology) with the extension of a new concept, since he/she has not a formal definition of the concept, the place of the ontology in which it has to be inserted is not specified. This fact typically occurs in settings where ontology engineer detects language poorness in the ontology. This point of view gives rise to *user-driven* approaches that we have formalized in [2].

The remainder of the paper is organized as follows. In the next section we present a simple example to illustrate the problem. In section 3 the formalization of *robust ontology extension* is outlined. A kind of extension is the extension by *insertion of an undefinition* (sect. 4). The method is applied to solve the problem of the example. Finally, some final remarks about the approach are given in section 5.

2 A Motivating Example

In order to understand the problem as well as its solution, let us suppose that a Geographical Information System (GIS) launches agents for finding, in the SW, information about several geographical objects in United States. Suppose that the data set Δ found by the agents is:

`Overlap`($West$, $Mount\ Elbert$)
`PartOf`($Mount\ Elbert$,$Colorado$)
`ProperPartOf`($Miami$, $Florida$)
`ProperPartInverse`($Florida$, $Miami$)
`PartialOverlaps`($Basin\ of\ Missouri\ River$,$West$)
`Overlaps`($Basin\ of\ Platte\ River$, $Nebraska$)
`TangentialProperPart`($Mount\ Elbert$, $GreatPlains$)
`Discrete`($Colorado$, $Basin\ of\ Missouri\ River$)
`PartOf`($Colorado$, $West$)
`ProperPart`($East$, $Colorado$)
`PartOf`($Miami$, $Florida$)
`Overlaps`($East$, $Miami$)
`Overlaps`($West$, $Colorado$)
`Discrete`($West$, $Georgia$)
`Part`($East$, $Georgia$)

Note that several facts do not provide the most specific spatial relation that it might be expressed by the ontology. That is the case of the fact `Overlaps`($Basin\ of\ Platte\ River$, $Nebraska$). Both regions are overlapping, however there is no information about what level of overlapping relates these regions. Since the GIS deals with concepts representing underspecify spatial relations such as `Overlaps`, or `PartOf, ...`, it is hard to classify individual regions in an accurate way. They would be classified to work within a set of specific spatial-relations/concepts, a jointly exhaustive set of pairwise disjoint (JEPD) concepts to get the exhaustive intended classification.

The problem can be stated as: *Given a set Δ of facts with respect to an ontology O, where the most specific information on some individuals can not*

entailed, to design an provisional robust extension of O to provisionally classify these concepts.

The ontology for the running example is *Region Connection Calculus* (RCC), designed for (mereotopological) Qualitative Spatial Reasoning (QSR)[7]. The relations of RCC are used in both GIS and spatial databases [9]. More information on RCC can be found in [7].

The jointly exhaustive and pairwise disjoint (JEPD) set of binary relations depicted in figure 1 (right-bottom) is denoted by RCC8. The complexity of RCC8 to solve Constraints Satisfaction Problems (CSP) has been deeply studied by J.R. Renz and B. Nebel [11]. Other calculus to take into account is RCC5. It is based on the set $\{DR, PO, PP, PPi, EQ\}$. It is less precise but more manageable than RCC8. Therefore, RCC8 represents the most specific spatial relationships in RCC. The remaining relations of RCC can be regarded as *unprecise*. The special interest of authors in this ontology lies in its role as meta-ontology for visual cleaning [4].

3 Extending Ontologies with Backward Compatibility

The study of ontology revision covers a very broad spectrum of theories and techniques. It encompasses logical and engineering methods, including theories from the fields of KDE and Knowledge Representation and Reasoning. A typical case of the need of ontology revision occurs when ontology engineer detects that new data are not accurately specified/classified with respect to the current information. A first solution may be to insert some provisional concept(s) (*notion(s)*) classifying that unprecise information and to expect that new conditions will allow us to refine information for obtaining a right classification. Actually, it involves an *extension* of ontology. For instance the existence of ground literals (instances) of an abstract concept (i.e. they are non-direct instances) can be a methodological problem in ontology design. Thus, it is better to consider a new concept that provisionally represents a notion. As we have already commented, such a concept will not have subclasses; thus, it will be located at the ground level of the taxonomy of concepts.

It is necessary to point out that ontology evolution must obey basic accepted principles such as *backward compatibility*, while it is possible. In [3] a weak form of backward compatibility, useful for the aim of this paper, is introduced. In fact, it has been used for other kind of extensions in [2].

Considering an ontology as a pair (T, E) where T is the axioms set and E is a equational characterization of the intended lattice of concepts, (the *skeleton*), we say that an ontology is *lattice categorical* (l.c.) if the lattice determined by T, and denoted by $L(T, \mathcal{C})$, is unique. $\mathcal{C}$ denotes the set of concepts of T. The theory RCC is an example of l.c. theory. The only possible lattice structure exhibited by the models of RCC is that of figure 1, and a skeleton E is computed in [3].

In [3] and [2] we replaced *completeness* by *lattice categoricity* to facilitate the design of feasible methods for extending ontologies with logical soundness. The extension is defined as follows. Given $(T_1, E_1), (T_2, E_2)$, two ontologies of this

$$
\begin{aligned}
\top &\equiv C \sqcup DR \\
PO &\sqsubseteq \neg P \sqcap \neg Pi \sqcap \neg DR \\
DR &\equiv EC \sqcup DC \\
NTPP &\sqsubseteq \neg TPP \sqcap \neg Pi \sqcap \neg DR \\
C &\equiv O \sqcup EC \\
TPP &\sqsubseteq \neg Pi \sqcap \neg DR \\
O &\equiv PO \sqcup P \sqcup Pi \\
EQ &\sqsubseteq \neg PPi \sqcap \neg DR \\
Pi &\equiv EQ \sqcup PPi \\
TPPi &\sqsubseteq \neg NTPPi \sqcap \neg DR \\
P &\equiv EQ \sqcup PP \\
NTPPi &\sqsubseteq \neg DR \\
PPi &\equiv TPPi \sqcup NTPPi \\
EC &\sqsubseteq \neg DC \\
PP &\equiv TPP \sqcup NTPP
\end{aligned}
$$

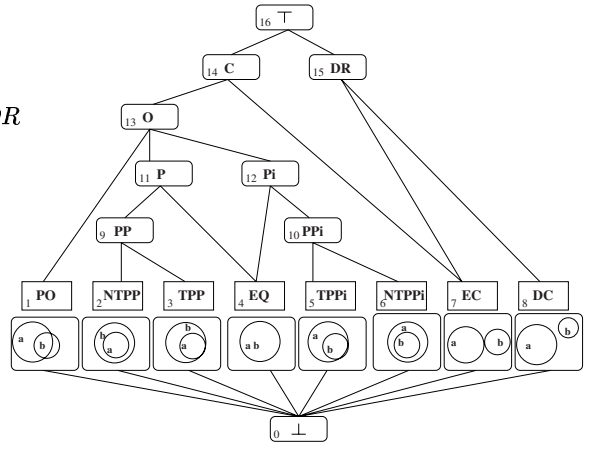


Fig. 1. The skeleton E (left) for the lattice of RCC (right)

kind with respect to the sets of concepts $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively, we say that (T_2, E_2) is a **lattice categorical extension** of (T_1, E_1) if $L(T_1, \mathcal{C}_1) \subseteq L(T_2, \mathcal{C}_2)$ and $L(T_2, \mathcal{C}_2) \models E_1$.

3.1 Cognitive Support

Once the notion of *lattice categorical extension* has been introduced, some functions for selecting the best l.c. extension have to be designed.

Suppose that $\Delta = \{h_1, \ldots h_n\}$ is a set of facts on concepts in $\mathcal{C}$. The user aims to classify some of individuals appearing in Δ by means of specific concepts. We can suppose, to simplify the notation, that every fact explicit in T belongs to Δ.

The **cognitive support** of C with respect to Δ, T and L, is

$$sup^{L}_{T,\Delta}(C) := \frac{|\{\boldsymbol{a} \in U(\Delta) \ : \ \exists i[C_i \leq C \wedge T \cup \Delta \models C_i(\boldsymbol{a})]\}|}{|U(\Delta)|}$$

where $U(\Delta) := \{\boldsymbol{a} \ : \ \text{exists } C \in \mathcal{C} \ [C(\boldsymbol{a}) \in \Delta]\}$ is the universe determined by Δ. That is, the cognitive support estimates the number of facts on the concept C that T entails (normalized by the size of $U(\Delta)$). The computation is trivial for lattice categorical theories, [2]: $sup^{L}_{T,\Delta}(C) = \dfrac{|C|^{\Delta}_{T}}{|U(\Delta)|}$ where $|C|^{\Delta} := |\{\boldsymbol{a} \ : C(\boldsymbol{a}) \in \Delta\}|$ and $|C|^{\Delta}_{T} := |\{\boldsymbol{a} \in U(\Delta) \ : \ T \cup \Delta \models C(\boldsymbol{a})\}|$.

Suppose now that Δ is compounded by facts on atoms of the lattice of concepts (that is, about the most specific concepts). In this case, since $\mathcal{J} = \{C_1, \ldots, C_n\}$ is a JEPD, $sup_{T,\Delta}(.)$ is a probability measure. In general, if $\mathcal{J}$ is a JEPD set of concepts in L, and Δ is compounded by instances on concepts falling in the cone of some element of $\mathcal{J}$, then $sup_{T,\Delta}(.)$ is a probability measure on $\mathcal{J}$.

Finally, the **cognitive entropy** of $\mathcal{J}$ is

$$CH(\mathcal{J}) = -\sum_{C \in \mathcal{J}} sup_{T,\Delta}(C) \log sup_{T,\Delta}(C)$$

This entropy is the key parameter used in the *user-driven* approach [2].

4 Data-Driven Ontology Revision for Defficent Data

A defficent classification of data induces the insertion of subconcepts for refining the classification of individuals which initially were misclassified. As it is already commented, the new concepts will fall in the bottom level. Therefore, we aim to extend $\mathcal{J}_L$, the JEPD set of concepts which are the atoms of the lattice $L(T, \mathcal{C})$.

The following definition formalizes the notion of *insertion of a concept with certain degree of unprecision* as subconcept of a given concept C. It has to be determined whether there is a l.c. extension of the ontology with an (atomic) subconcept μC of C. Intuitively, the meaning of $\mu C(\mathbf{a})$ is "the concept $\mathbf{a}$ falls in the concept C, but we do not know more specific information about $\mathbf{a}$". Formally,

Definition 1. *Let (T, E_0) be an ontology and $C \in \mathcal{C}$. We say that the ontology admits an undefinition at C ($T \rightsquigarrow_w C$) if there is a l.c. extension of T, (T', E'), such that*

1. *T' is l.c. with respect to $\mathcal{C} \cup \{\mu C\}$, (where $\mu C \notin \mathcal{C}$).*
2. *$\{\mu C\}$ is an atom in the lattice $L' = L(T, \mathcal{C} \cup \{\mu C\})$.*
3. *There is not C' such that $\mu C <^{L'} C' <^{L'} C$.*

Note that, in above conditions, $\mathcal{J}_L[\mu C] := \mathcal{J}_L \cup \{\mu C\}$ is a JEPD set for L' (see fig. 2, left). This requirement represents, in fact, that we have not any additional information about μC. For example, in figure 2 right, the relation $\mu C(a, b)$ means "the regions a and b are connected, but it is unknown if they overlap or they are externally connected".

The notation $T \models_\mu C(\boldsymbol{a})$ means $T \models C(\boldsymbol{a})$ and, for all $D <^L C$, $T \not\models D(a)$. In other words, $C(a)$ is the most specific knowledge on $\mathbf{a}$ entailed by T. It is easy to see that

Proposition 1. *Whatever two extensions of T by undefinition at C have equivalent lattice skeletons modulo completion.*

Such a skeleton of the extension is denoted by $E[\mu C]$. We also consider the iteration of this kind of extensions, namely $E[\mu C_1, \ldots, \mu C_k]$.

Corollary 1. *$E[\{\mu C : C \in \mathcal{C} \wedge T \rightsquigarrow_w C\}]$ is unique (modulo database completion axioms).*

For the example, this kind of extensions for RCC have to be investigate.

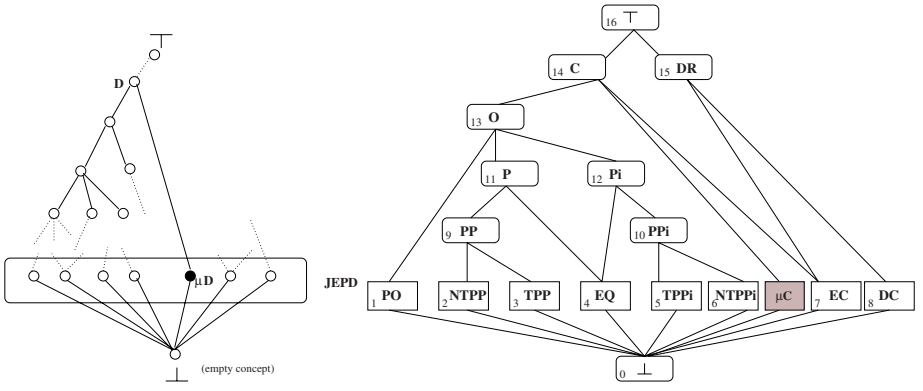


Fig. 2. The ontology admits an undefinition in the concept C (*connection*) (right)

4.1 Inserting Provisional Spatial Relationships in RCC

As we have already commented, the JEPD set named RCC8 is the representation of a precise classification for RCC.

Theorem 1. *There are exactly eight extensions by undefinition of the lattice of RCC by insertion of a new relation D such that RCC8 $\cup \{D\}$ is a JEPD set.*

Such new relations can be mereotopologically interpreted [6]. The lattices of the extensions are detailed at [3]. For example, the lattice depicted in fig. 1 (right) has a skeleton $E[\mu C]$.

The next step consists in deciding which is the best l.c. extension to classify data. Suppose that $\Delta = \{h_1, \ldots h_n\}$ is the set of facts. Assume that the user believes that the set of misclassified elements is $I = \{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_k\} \subseteq U(\Delta)$ (according with user's ontology). In this case, the problem is not due to a new concept, because he/she has not decided yet an insertion. Such elements are not falling on atomic concepts ($T \not\models C(\boldsymbol{a})$ for any $C \in \mathcal{J}_L$), because the user has not an specific definition of them, that is, he has got only unprecise information (as instances of upper concepts).

It is easy to provide an extension by undefinition with complete classification of data. For each $\boldsymbol{a}_i \in I$, let $C^i \in \mathcal{C}$ such that $T \models_\mu C^i(\boldsymbol{a}_i)$. Any extension by undefinition at the set $\{C^i \ : \ i = 1, \ldots k\}$ classifies every element of $U(\Delta)$ with a concept of the JEPD set $\mathcal{J}_{T'} := \mathcal{J} \cup \{\mu C^1, \ldots, \mu C^k\}$. Note also, that if we do not require C^i is the most specific one, the extension is not unique.

Definition 2. *Let T' be an extension by undefinition of T as defined in 1. The support of μC is defined as*

$$supp_{T',\Delta}(\mu C) = \frac{|\{\boldsymbol{a} \in U(\Delta) : \boldsymbol{a} \in I \wedge T \cup \Delta \models_\mu C(\boldsymbol{a})\}|}{|U(\Delta)|}$$

That is, the support of μC uses the number of elements for such that T proves they belong to C. In this way $supp_{T',\Delta}$ is also a probability measure on $\mathcal{J}_{T'}$.

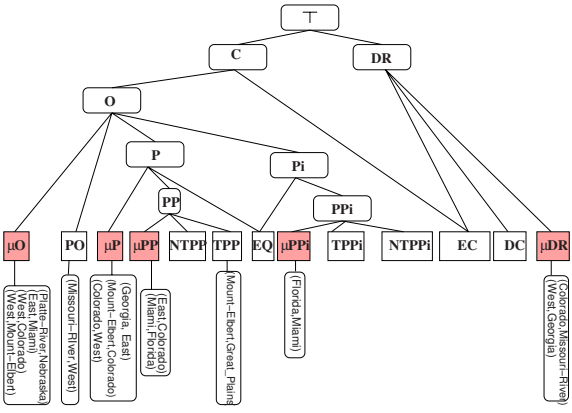


Fig. 3. Classification of data according to $E[\mu PP, \mu P, \mu PPi, \mu O, \mu DR]$

Note that this computation is equivalent to consider the support with respect to the theory $T' + \{\mu C(\boldsymbol{a}) \ : \ T \cup \Delta \models_\mu C(\boldsymbol{a})\}$. To simplify the notation, we finally consider throughout that T' is that theory.

Theorem 2. *The extension above defined exhibits the maximum cognitive entropy among every possible extension by undefinition classifying* $U(\Delta)$.

Sketch of proof: If T'' is other extension, then some $\boldsymbol{a}_i$ of I are classified with respect to a concept which is not the most specific one w.r.t. T. Thus, the result follows by the convexity of the function $p \log p$.

A l.c. extension by undefinition with maximum entropy gives little information on new concepts. This option is a *cautious* solution to the problem, because strong requirements for the new concepts are not been imposed.

The extension of RCC for the running example will be a combination of some of the eight extensions. We are interested to find an extension by undefinition of RCC that classifies the data and exhibits higher entropy. According to data and th. 2, the selected extension has skeleton (fig. 3): $E[\mu PP, \mu P, \mu PPi, \mu O, \mu DR]$. This l.c. extension has maximum entropy (by above theorem), 1.566. For example, $E[\mu P, \mu PPi, \mu O, \mu DR]$, shows entropy 1.326.

5 Closing Remarks and Related Work

A formalization of integration of unprecise data with respect to an ontology has been investigated, as well as a method to insert new concepts in an ontology with backward compatibility and preserving a weak form of completeness.

Note that reasoning services -that we need in order to build the extension with maximum entropy- can be non-decidable for first order theories. However, it is feasible for ontologies expressed in several (decidable) Description Logics, or considering the skeleton (a DL theory) as basis theory.

In [2] we formalize the insertion of a concept (possibly in a upper level) that will remain well defined once the appropriate extension is selected. In that case, the computation of the (conditional) entropies is easier than the entropies defined on this paper. The approach of this paper is different because it is not necessary user decision on new concepts. Possibly, both procedures should be combined in several situations like document enrichment tasks [10].

Entropy is usually considered for associating data and concepts of an ontology (see e.g. [5]). J. Calmet and A. Daemi also use entropy for revising or comparing ontologies [8], based on the concept taxonomy. However it is unusual to consider the *provability* as a parameter.

Finally, note that, although the method is fully formalized, the cognitive soundness of the extensions will depend of the human decision. Moreover, the iteration of the method can produce the existence of many provisional concepts without intentional component. It may be unadvisable in some cases.

References

1. Alonso-Jiménez, J.A., Borrego-Dáaz, J., Chávez-González, A.M., Martńn-Mateos, F.J.: Foundational Challenges in Automated Data and Ontology Cleaning in the Semantic Web. IEEE Intelligent Systems 21(1), 42–52 (2006)
2. Borrego-Díaz, J., Chávez-González, A.M.: Controlling Ontology Extension by Uncertain Concepts Through Cognitive Entropy. Uncertain Reasoning for the Semantic Web, URSW 2005, CEUR 173, 56–66 (2005)
3. Borrego-Díaz, J., Chávez-González, A.M.: Extension of Ontologies Assisted by Automated Reasoning Systems. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 253–257. Springer, Heidelberg (2005)
4. Borrego-Díaz, J., Chávez-González, A.M.: Visual Ontology Cleaning: Cognitive Principles and Applicability. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, pp. 317–331. Springer, Heidelberg (2006)
5. Brewster, C., Alani, H., Dasmahapatra, S., Wilks, Y.: Data Driven Ontology Evaluation, Int. Conf. Lang. Resources and Evaluation (2004), `http://eprints.ecs.soton.ac.uk/archive/00009062/01/BrewsterLREC-final.pdf`
6. Chávez-González, A.: Mereotopological Automated Reasoning for Ontology Cleaning, Ph.D. Thesis, University of Seville (2005)
7. Cohn, A.G., Bennett, B., Gooday, J.M., Gotts, N.M.: Representing and Reasoning with Qualitative Spatial Relations about Regions. In: Stock, O. (ed.) Spatial and Temporal Reasoning, ch. 4, Kluwer, Dordrecth (1997)
8. Daemi, A., Calmet, J.: From Ontologies to Trust through Entropy. In: Proc. of the Int. Conf. on Advances in Intelligent Systems - Theory and Applications (2004)
9. Grohe, M., Segoufin, L.: On First-Order Topological Queries. ACM Trans. Comput. Log. 3(3), 336–358 (2002)
10. Motta, E., Buckingham, S., Domingue, J.: Ontology-driven document enrichment: principles, tools and applications. Int. J. Human-Computer Studies 52(6), 1071–1109 (2000)
11. Renz, J.R., Nebel, B.: On the Complexity of Qualitative Spatial Reasoning: A Maximal Tractable Fragment of the Region Connection Calculus. Artificial Intelligence 108, 69–128 (1999)

Using Temporal Logic for Spatial Reasoning: Temporalized Propositional Neighborhood Logic

Antonio Morales, Isabel Navarrete, and Guido Sciavicco

Dept. of Information Engineering and Communications,
Faculty of Informatics, University of Murcia
30100, Espinardo (Murcia), Spain
{morales,inava,guido}@um.es

Abstract. In this paper we develop a new modal logic for spatial reasoning called Temporalized Propositional Neighborhood Logic, which is the result of the application of a known technique called temporalization to a known temporal logic for spatial reasoning (PNL). We will show that our logic is expressive enough to formalize interesting spatial statements and that it is decidable in NEXPTIME.

1 Introduction

It is widely accepted that spatial reasoning plays a central role in artificial intelligence, thanks to a variety of potential applications e.g. in robotics, geographical information systems, medical analysis and diagnosis. As for other qualitative reasoning formalisms (e.g., temporal reasoning), spatial reasoning can be viewed under three different, somehow complementary, points of view. We may distinguish between the *algebraic* level, that is, purely existential theories formulated as constraint satisfaction systems over jointly exclusive and mutually disjoint set of topological, directional, or combined relations; the *first-order* level, that is, first-order theories of topological, directional, or combined relations; and the *modal logic* level, where a (usually propositional) modal language is interpreted over opportune Kripke structures representing space. In this paper we develop a new modal logic for spatial reasoning called Temporalized Propositional Neighborhood Logic, which is the result of the application of a technique called temporalization [FG96] to a known temporal logic for spatial reasoning (PNL). We will show that our logic is expressive enough to formalize interesting spatial statements and that it is decidable in NEXPTIME.

2 Modal Logics for Spatial Reasoning

As for modal logics for spatial reasoning, it is worth mentioning Bennett's work [Ben94, Ben96], later extended by Bennett, Cohn, Wolter and Zakharyaschev in [BCWZ02]. In [Ben94], Bennett proposes to interpret regions as subsets of the topological space, and shows how it is possible to exploit both the classical propositional calculus and the intuitionistic propositional calculus, together

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 313–320, 2007.
© Springer-Verlag Berlin Heidelberg 2007

with certain meta-level constraint concerning entailments between formulas, for reasoning about space with topological relations. In [Ben96] Bennett extends his approach by the use of modal languages. In [BCWZ02], the authors consider a multi-modal system for spatio-temporal reasoning, based on Bennett's previous work; further research on this issue can be found in [Nut99]. Focusing on modal logics for spatial reasoning that exploit the whole expressive power of modal logic for reasoning about space (instead of using it for constraint solving), we mention here Lutz and Wolter's modal logic for topological relations [LW04], Venema's Compass Logic introduced in [Ven90] and further studied in [MR99], and Morales and Sciavicco's Spatial Propositional Neighborhood Logic [MS06b, MS06a]. Lutz and Wolter present a propositional modal logic, where propositional variables are interpreted in the regions of topological space, and references to other regions are enabled by modal operators interpreted as topological relations. Between the many possible choices for the set or relations, the logic for topological relations has been studied for the set RCC8 and for its refinement RCC5; the satisfiability problem for the modal logic of RCC8 relations interpreted in the Euclidean space $\mathbb{R}^n$ $(n > 1)$, when the set of basic regions is exactly the set of all (hyper)-rectangles on it, has been shown to be not even recursively enumerable, which means that it is not possible even to devise a semi-decidability method for it. As for directional relations, Venema's Compass Logic features four modal operators, namely $\diamondsuit\!\!\!\rightarrow$, $\diamondsuit\!\!\!\leftarrow$, $\diamondsuit\!\!\!\uparrow$, and $\diamondsuit\!\!\!\downarrow$, and propositional variables are interpreted as points in the Euclidean two-dimensional space. In [MR99], Marx and Reynolds show that Compass Logic is undecidable even in the class of all two-dimensional frames.

In [MS06a, MS06b] it has been presented a new modal logic for spatial reasoning called Spatial Propositional Neighborhood Logic (SpPNL for short). In SpPNL, regions are approximated by their minimum bounding box, and four modal operators allow one to move along the x- and the y-axis. SpPNL has been shown to be very expressive and useful for formalizing natural language spatial expressions. Unfortunately, its satisfiability problem is not decidable, and, thus, a natural question is whether exists or not some expressive enough, yet decidable, fragment of it. In this paper we consider a known technique called *temporalization* [FG96], for generating well-behaved multi-dimensional modal logics. While, so far, temporalization has been used on classical modal logics and on point-based temporal logics, we apply it on Propositional Neighborhood Logic [GMS03, BMGS07] (PNL), which is a decidable interval-based temporal logic. By temporalizing PNL with itself, we obtain a bi-dimensional interval temporal logic (SpPNL_t) which happens to be a decidable fragment of SpPNL. In this paper we study SpPNL_t, we show its relationship with SpPNL, and we prove its decidability.

3 A Simple Spatial Problem and Motivations

In the literature of qualitative spatial reasoning there exists no decidable logical formalism which is expressive enough to translate simple natural language statements such as the following one: *If at the north of the current region there*

is a region (whose width is equal to that of the current region) satisfying the property A, and the south of the current region there is no region satisfying the property B, then at the east of the current region there must be a region satisfying the property C. The logic developed in this paper is interesting from two different points of view: (1) so far, the technique of temporalization (more in general, combination of logics) has been applied to modal logics and point-based temporal logics; the interesting results obtained in such cases naturally arise the question of wether combining interval based temporal logics allows one to draw similar conclusions concerning transferring positive results to the combined logic from the initial one(s); (2) although the modal logic we will present here is not especially expressive, it allows one to formalize statements such the one above, and since it is decidable, one might directly apply it in the context of spatial databases and qualitative spatial reasoning.

4 Background

4.1 Propositional Neighborhood Logic

Let $\mathbb{D} = \langle D, < \rangle$ be a linearly ordered set. An *interval* over $\mathbb{D}$ is an ordered pair $[d_0, d_1]$, where $d_0, d_1 \in D$ and $d_0 < d_1$. We denote the set of intervals over $\mathbb{D}$ by $\mathbb{I}(\mathbb{D})$. The language of *Propositional Neighborhood Logic* (PNL) consists of a set $\mathcal{AP}$ of propositional letters, the propositional connectives $\neg$ and $\vee$, and the modal operators $\langle A \rangle$ and $\langle \overline{A} \rangle$. The other propositional connectives, as well as the logical constants $\top$ (*true*) and $\bot$ ($false$) and the dual modal operators $[A]$ and $[\overline{A}]$, are defined as usual. *Formulas* of PNL, denoted by $f, g, \ldots$, are recursively defined by the following grammar:

$$f ::= p \mid \neg f \mid f \vee g \mid \langle A \rangle f \mid \langle \overline{A} \rangle f$$

The semantics of PNL is given in terms of *interval models* $M = \langle \mathbb{I}(\mathbb{D}), V \rangle$. The *valuation function* $V : \mathcal{AP} \mapsto 2^{\mathbb{I}(\mathbb{D})}$ assigns to every propositional variable p the set of intervals $V(p)$ over which p holds. As a matter of fact, for every p, $V(p)$ can be viewed as a binary relation on D. The *truth relation* for a formula and a given interval in a model M is defined by structural induction on formulas; the modal clauses are:

(1) $M, [d_0, d_1] \Vdash \langle A \rangle f$ iff there exists d_2 s.t. $d_1 < d_2$ and $M, [d_1, d_2] \Vdash f$;
(2) $M, [d_0, d_1] \Vdash \langle \overline{A} \rangle f$ iff there exists d_2 s.t. $d_2 < d_0$ and $M, [d_2, d_0] \Vdash f$.

A formula is *satisfiable* if it is true over some interval in some interval model (for the respective language) and it is *valid* if it is true over every interval in every interval model. As shown in [GMS03], PNL is powerful enough to express interesting temporal properties, e.g., they allow one to constrain the structure of the underlying linear ordering. In particular, it allows one to express the *difference* operator, the *universal* modality, and thus to simulate *nominals*. This properties will be useful for the rest of this paper.

Theorem 1. *[BMGS07, BMS] The satisfiability problem for PNL interpreted in the class of all linearly ordered sets is decidable in NEXPTIME.*

4.2 Spatial Propositional Neighborhood Logic

The language for SpPNL ([MS06b]) consists of a set of propositional variables $\mathcal{AP}$, the logical connectives $\neg$ and $\vee$, and the modalities $\langle \mathrm{E} \rangle, \langle \mathrm{W} \rangle, \langle \mathrm{N} \rangle, \langle \mathrm{S} \rangle$. The other logical connectives, as well as the logical constants $\top$ and $\bot$, can be defined in the usual way. SpPNL well formed *formulas*, denoted by $\phi, \psi, \ldots$, are recursively defined as follows (where $p \in \mathcal{AP}$):

$$\phi = p \mid \neg\phi \mid \phi \vee \psi \mid \langle \mathrm{E} \rangle \phi \mid \langle \mathrm{W} \rangle \phi \mid \langle \mathrm{N} \rangle \phi \mid \langle \mathrm{S} \rangle \phi.$$

Given any two linearly ordered sets $\mathbb{H} = \langle H, < \rangle$ and $\mathbb{V} = \langle V, < \rangle$, we call *spatial frame* the structure $\mathbb{F} = (\mathbb{H} \times \mathbb{V})$, and we denote by $\mathbb{O}(\mathbb{F})$ the set of all *objects* (rectangles), that is, $\mathbb{O}(\mathbb{F}) = \{\langle (h, v), (h', v') \rangle \mid h < h' \ v < v', \ h, h' \in \mathbb{H} \ v, v' \in \mathbb{V}\}$. The semantics of SpPNL is given in terms of *spatial models* of the type $M = \langle \mathbb{F}, \mathbb{O}(\mathbb{F}), \mathcal{V} \rangle$, where $\mathbb{F}$ is a spatial frame, and $\mathcal{V} : \mathbb{O}(\mathbb{F}) \mapsto 2^{\mathcal{AP}}$ is a *spatial valuation function*. The *truth* relation for a well formed SpPNL-formula ϕ in a model M and an object $\langle (h, v), (h', v') \rangle$ is defined by structural induction; the modal clauses are:

(1) $M, \langle (h, v), (h', v') \rangle \Vdash \langle \mathrm{E} \rangle \phi$ if only if there exists $h'' \in \mathbb{H}$ such that $h' < h''$, and $M, \langle (h', v), (h'', v') \rangle \Vdash \phi$;
(2) $M, \langle (h, v), (h', v') \rangle \Vdash \langle \mathrm{W} \rangle \phi$ if only if there exists $h'' \in \mathbb{H}$ such that $h'' < h$, and $M, \langle (h'', v), (h, v') \rangle \Vdash \phi$;
(3) $M, \langle (h, v), (h', v') \rangle \Vdash \langle \mathrm{N} \rangle \phi$ if only if there exists $v'' \in \mathbb{V}$ such that $v' < v''$, and $M, \langle (h, v'), (h', v'') \rangle \Vdash \phi$;
(4) $M, \langle (h, v), (h', v') \rangle \Vdash \langle \mathrm{S} \rangle \phi$ if only if there exists $v'' \in \mathbb{V}$ such that $v'' < v$, and $M, \langle (h, v''), (h', v) \rangle \Vdash \phi$;

As usual, we denote by $[\mathrm{X}]$ the dual operator of the modality $\langle \mathrm{X} \rangle$, where $\langle \mathrm{X} \rangle \in \{\langle \mathrm{E} \rangle, \langle \mathrm{W} \rangle, \langle \mathrm{N} \rangle, \langle \mathrm{S} \rangle\}$, and by $M \Vdash \phi$ the fact that ϕ is *valid* on M.

In [MS06a] it has been shown that SpPNL is very expressive, despite its simplicity; nevertheless its satisfiability problem is not decidable. In [MNS07] it has been presented a sound and complete axiomatic system for SpPNL, and a non-terminating tableaux procedure for semi-decidability; moreover, it has been posed the following question:

Question 1: is it possible to find out some kind of syntactically defined sub-logic of SpPNL whose satisfiability/validity problem is decidable?

5 Temporalizing Propositional Neighborhood Logic

We now consider the problem of temporalizing PNL. The technique called *temporalization* [FG96] is considered the simplest form of combination of two (temporal) logics, and, in the case of linear time point-based temporal logics, it allows one to transfer positive results, such as completeness and decidability of two (temporal) logics, to the combined one. As far as we know, in the literature there is no previous attempt of temporalizing interval-based temporal logics.

Since we want to combine PNL with itself, we will denote by $\langle A\rangle_h$ and $\langle\overline{A}\rangle_h$ the modal operators from PNL used in the 'horizontal' domain, and by $\langle A\rangle_v$ and $\langle\overline{A}\rangle_v$ those used in the 'vertical' domain. One of our objectives is to show that the combined logic PNL(PNL) is a (strict) sublogic of SpPNL; thus, we will denote it by SpPNL$_t$ (where t stands for 'temporalized'). Formulas of SpPNL$_t$ will be denoted by $\phi, \psi \ldots$ (since, as we will show, they are also SpPNL-formulas), and are generated by the following abstract syntax:

$$\phi ::= f \mid \neg\phi \mid \psi \vee \varphi \mid \langle A\rangle_h\phi \mid \langle\overline{A}\rangle_h\phi,$$

where f is a usual PNL-formulas such that every occurrence of the modal operators $\langle A\rangle$ y $\langle\overline{A}\rangle$ has been replaced by $\langle A\rangle_v$ and $\langle\overline{A}\rangle_v$, respectively. Intuitively, a SpPNL$_t$-formula is a PNL-formula where, instead of propositional letter, we use again PNL-formulas; by the outermost modal operators we move, so to say, horizontally on a linear domain, while by the innermost modal operators we move vertically. The temporalization method allows the two component logics to interact in a very restricted way, that is, first one moves using one logic, and, then, the other one. As for the semantics, we consider a linearly ordered set $\mathbb{H} = \langle H, <\rangle$ such that, for any interval $[h_i, h_j]$ $(h_i < h_j)$ from the set $\mathbb{I}(\mathbb{H})$ a *partial evaluation function* $\nu : \mathbb{I}(\mathbb{H}) \mapsto \mathfrak{M}$ (where by $\mathfrak{M}$ we denote the infinite set of all possible PNL-models) it is defined as the PNL-model $\nu([h_i, h_j]) = \langle \mathbb{V}^{[h_i,h_j]}, \mathbb{I}(\mathbb{V}^{[h_i,h_j]}), \mathcal{V}^{[h_i,h_j]}\rangle$, where $\mathbb{V}^{[h_i,h_j]} = \langle V^{[h_i,h_j]}, <\rangle$ is a linearly ordered set, in which we identify a point $v_0^{[h_i,h_j]}$. Thus, a SpPNL$_t$-model is a tuple $\mathrm{M} = \langle \mathbb{H}, \mathbb{I}(\mathbb{H}), \nu\rangle$. Given a SpPNL$_t$-model M, an interval $[h_i, h_j]$ on it, and a SpPNL$_t$-formula ϕ, the *truth* relation is defined as follows:

(1) $M, [h_i, h_j] \Vdash f$, where f is a PNL-formula, iff for some v_k such that v_k belongs to the domain of $\nu([h_i, h_j])$ and $v_0 < v_k$, the PNL-model $N = \nu([h_i, h_j])$ is such that $N, [v_0^{[h_i,h_j]}, v_k^{[h_i,h_j]}] \Vdash f$;
(2) $M, [h_i, h_j] \Vdash \neg\psi$ iff it is not the case that $M, [h_i, h_j] \Vdash \psi$;
(3) $M, [h_i, h_j] \Vdash \phi \vee \psi$ iff $M, [h_i, h_j] \Vdash \phi$ or $M, [h_i, h_j] \Vdash \psi$;
(4) $M, [h_i, h_j] \Vdash \langle A\rangle_h\psi$ iff there exists h_k such that $h_j < h_k$ and $M, [h_j, h_k] \Vdash \psi$;
(5) $M, [h_i, h_j] \Vdash \langle\overline{A}\rangle_h\psi$ iff there exists h_k such that $h_k < h_i$ and $M, [h_i, h_j] \Vdash \psi$.

As for example, the following one is a SpPNL$_t$-formula: $p \wedge \langle A\rangle_h(q \vee [A]_v\neg p)$, while the following one is not: $\langle A\rangle_v\langle A\rangle_h(p \wedge \langle A\rangle_h\langle A\rangle_v q)$.

6 Temporalized PNL as a Fragment of SpPNL

In this section, we want to show that SpPNL$_t$ is a proper fragment of SpPNL. For sake of simplicity we restrict our attention to frames built up from countable linearly ordered sets and frames.

Theorem 2. *On countable frames, it holds that* SpPNL$_t \subset$ SpPNL.

Proof
[sketch] Due to space limitations, details are omitted. First observe that the relation SpPNL $\subset$ SpPNL$_t$ is not possible, since the abstract grammar SpPNL$_t$ does not even allow to construct some valid SpPNL formulas; thus, we have that at least SpPNL$_t \neq$ SpPNL. In order to prove the rest of the theorem, we need a satisfiability-preserving transformation ξ for both models and formula. Thus, let $\mathrm{M} = \langle \mathbb{H}, \mathbb{I}(\mathbb{H}), \nu \rangle$ be any SpPNL$_t$-model, and define the SpPNL-model $\xi(\mathrm{M}) = \langle \mathbb{F}, \mathbb{O}(\mathbb{F}), \mathcal{V} \rangle$ as follows: (i) $\mathbb{F} = \mathbb{H} \times \mathbb{V}$, where $\mathbb{V}$ is any countable linearly ordered set isomorphic to $\mathbb{V}^{[h^*,h^{*'}]}$, being $\mathbb{V}^{[h^*,h^{*'}]}$ a set of vertical components of M of maximal cardinality; (ii) $\mathcal{V}$ is defined in such a way that for each propositional letter p, $p \in \mathcal{V}(\langle (h, v_i), (h', v_j) \rangle)$ if and only if $p \in \mathcal{V}^{[h,h']}([v_i, v_j])$ where $[v_i, v_j] \in \mathbb{I}(\mathbb{V})$ (v_i and v_j are the i-th and j-th elements of $\mathbb{V}$) and $\mathbb{V}$ is the linear domain of $\nu([h, h'])$. It is possible to show that ξ can be extended to formulas, and that, for any SpPNL$_t$-formula ϕ, ϕ is satisfiable if and only if $\xi(\phi)$ is satisfiable. In particular, it is possible to show by double induction that, if $\mathrm{M} = \langle \mathbb{H}, \mathbb{I}(\mathbb{H}), \nu \rangle$ is a SpPNL$_t$-model, then $\mathrm{M}, [h, h'] \Vdash \phi$ if and only if there exists an object $\langle (h, v)(h', v') \rangle$ such that $\xi(\mathrm{M}), \langle (h, v), (h', v') \rangle \Vdash \xi(\phi)$. ■

7 Decidability and Expressive Power of SpPNL$_t$

In this section, we show that the satisfiability problem for SpPNL$_t$ is decidable; to this end, we will use the result of Theorem 1. Borrowing some of the terminology from [FG96], we define SpPNL$_t$ monolithic formulas as follows.

Definition 3. *Let $\phi \in$ SpPNL$_t$. Any sub-formula ψ of ϕ is said to be* monolithic *if and only if it is a classical propositional formula, or it is a* SpPNL$_t$*-formula such that the outermost operator is one of $\{\langle A \rangle_v, \langle \overline{A} \rangle_v, [A]_v, [\overline{A}]_v\}$.*

For example, $p \wedge \langle A \rangle_v p$ and $\langle A \rangle_v (p \vee q)$ are monolithic formulas, while $\langle A \rangle_h \langle A \rangle_v p$ is not. Clearly, monolithic formulas are standard PNL-formulas.

Theorem 4. *The satisfiability problem for* SpPNL$_t$ *interpreted in the class of all frames (resp., discrete frames) is decidable in NEXPTIME.*

Proof
[sketch] The idea is that monolithic formulas can be 'simulated' by propositional letters taken from a vocabulary $\mathcal{AP}'$ such that $\mathcal{AP}' \cap \mathcal{AP} = \emptyset$. We want to replace monolithic formulas with propositional letters, and then check the satisfiability of the resulting PNL-formula. Consider the following translation γ from SpPNL$_t$-formulas to PNL-formulas; let ϕ be a SpPNL$_t$-formula and, on the complexity of ϕ, define, $\gamma(f) = p'$, where $p' \in \mathcal{AP}'$ is a fresh propositional variable, and f is a maximal monolithic formula (in the sense that there is no other monolithic formula g that f is a sub-formula of g), and, then, extend γ to formulas by simply substituting each occurrence of $\langle A \rangle_h$ (resp., $\langle \overline{A} \rangle_h$) with $\langle A \rangle$ (resp., $\langle \overline{A} \rangle$). We first prove that, for any SpPNL$_t$-formula ϕ, ϕ is satisfiable if and only if a certain PNL-formula $\gamma^*(\phi)$ is satisfiable; then, we use the decidability result for

PNL. For any SpPNL$_t$-formula ϕ, the formula $\gamma(\phi)$ is already a PNL-formula, but, clearly, it is not equisatisfiable to ϕ; thus we have to modify it. The problem is that in $\gamma(\phi)$ the propositional letters that replace monolithic formulas are all distinct; we have to identify those monolithic formulas which have the same meaning (i.e., the same models), and formalize such a constraint in the language of PNL. Let $\mathcal{F}$ be the set of all maximal monolithic formulas in ϕ; for each pair $f, g \in \mathcal{F}$ (that is, $f, g \in$ PNL), we have that f, g are equisatisfiable if and only if the formula $f \leftrightarrow g$ is satisfiable in PNL, which is decidable (by Theorem 1). Now, let $[All]$ denote the universal operator in PNL, and let $f_1, f_2, \ldots, f_k$ be the set of all formulas of the type $f \leftrightarrow g$ above described. We have that $\gamma^*(\phi)$ is $\gamma(\phi) \wedge \bigwedge_{i=1}^{k} [All] f_i$. Clearly, a sound, complete and terminating decision algorithm is easily deducted. ■

Consider again the spatial statement of Section 3. Now, we are able to formalize it in the temporalized logic, as follows:

- *If at the north of the current region there is a region satisfying the property A:* $\langle A \rangle_v A \vee \langle A \rangle_v \langle A \rangle_v A$;
- *(If) at the south of the current region there is no region satisfying the property B:* $\neg \langle A \rangle_v B$;
- *Then at the east of the current region there must be a region satisfying the property C:* $\langle A \rangle_h C$,

which ends up in:

$$(\langle A \rangle_v A \vee \langle A \rangle_v \langle A \rangle_v A) \wedge \neg \langle A \rangle_v B \rightarrow \langle A \rangle_h C.$$

References

[BCWZ02] Bennett, B., Cohn, A.G., Wolter, F., Zakharyaschev, M.: Multi-dimensional modal logic as a framework for spatio-temporal reasoning. Applied Intelligence 17(3), 239–251 (2002)

[Ben94] Bennett, B.: Spatial reasoning with propositional logics. In: Doyle, J., Sandewall, E., Torasso, P. (eds.) KR'94: Principles of Knowledge Representation and Reasoning, pp. 51–62. Morgan Kaufmann, San Francisco, California (1994)

[Ben96] Bennett, B.: Modal logics for qualitative spatial reasoning. Journal of the Interest Group in Pure and Applied Logic (IGPL) 4(1), 23–45 (1996)

[BMGS07] Bresolin, D., Montanari, A., Goranko, V., Sciavicco, G.: On decidability and expressiveness of propositional neighborhood logics. In: Proc. of the Symposium on Logical Foundations of Computer Science 2007. LNCS, vol. 4514, pp. 84–99. Springer, Heidelberg (2007)

[BMS] Bresolin, D., Montanari, A., Sala, P.: An optimal tableau-based decision algorithm for propositional neighborhood logic. In: Thomas, W., Weil, P. (eds.) STACS 2007. LNCS, vol. 4393, Springer, Heidelberg (2007)

[FG96] Finger, M., Gabbay, D.: Combining temporal logic systems. Notre Dame Journal of Formal Logic 37(2), 204–232 (1996)

[GMS03] Goranko, V., Montanari, A., Sciavicco, G.: Propositional interval neighborhood temporal logics. Journal of Universal Computer Science 9(9), 1137–1167 (2003)

[LW04] Lutz, C., Wolter, F.: Modal logics of topological relations. In: Proc. of Advances in Modal Logics 2004 (2004)

[MNS07] Morales, A., Navarrete, I., Sciavicco, G.: Proof methods for spatial propositional neighborhood logic. In: Guesguen, H.W., Ligozat, G. (eds.) Proc. of the IJCAI-07 Workshop on Spatial and Temporal Reasoning (2007)

[MR99] Marx, M., Reynolds, M.: Undecidability of compass logic. Journal of Logic and Computation 9(6), 897–914 (1999)

[MS06a] Morales, A., Sciavicco, G.: Una comparativa entre el algebra de rectangulos y la logica SpPNL (in Spanish). In: Proc. of the Conference Campus Multidisciplinar en Perception e Inteligencia (CMPI-06), pp. 576–587 (2006)

[MS06b] Morales, A., Sciavicco, G.: Using time for spatial reasoning: Spatial propositional neighborhood logic. In: Proc. of the 13th International Symposium on Temporal Representation and Reasoning (TIME-06), pp. 50–57 (2006)

[Nut99] Nutt, W.: On the translation of qualitative spatial reasoning problems into modal logics. In: Burgard, W., Christaller, T., Cremers, A.B. (eds.) KI-99: Advances in Artificial Intelligence. LNCS (LNAI), vol. 1701, pp. 113–124. Springer, Heidelberg (1999)

[Ven90] Venema, Y.: Expressiveness and completeness of an interval tense logic. Notre Dame Journal of Formal Logic 31(4), 529–547 (1990)

An Integrated Approach to Filtering Phishing E-mails

M. Dolores del Castillo, Ángel Iglesias, and J. Ignacio Serrano

Instituto de Automática Industrial. CSIC,
Ctra. Campo Real, km. 0,200
28500 Arganda del Rey, Madrid, Spain
{lola,iglesias,nachosm}@iai.csic.es
http://www.iai.csic.es

Abstract. This paper presents a system for classifying e-mails into two categories, legitimate and fraudulent. This classifier system is based on the serial application of three filters: a Bayesian filter that classifies the textual content of e-mails, a rule based filter that classifies the non-grammatical content of e-mails and, finally, a filter based on an emulator of fictitious accesses which classifies the responses from websites referenced by links contained in e-mails. The approach of this system is hybrid, because it uses different classification methods, and also integrated, because it takes into account all kind of data and information contained in e-mails.

Keywords: e-mail attacks, textual and non-textual content, machine learning methods.

1 Filtering Methods

Spam has evolved to a new and dangerous form known as *phishing*. Phishing is the term used to describe massive emails that trick recipients into revealing their personal or company confidential information such as social security and financial account numbers, account passwords and other identity or security information.

Common attacks, which are signs of an electronic fraud, are e-mails that a user receives as if they were sent by a legitimate entity. These e-mails request the user's personal information as a client of this legitimate entity with a link to a website that looks like the legitimate entity's website or with a form contained in the body of the e-mail.

According to Anti-Phishing Working Group [9] the number of phishing reports increased from 20,109 in May 2006 to 28,571 in June 2006 to make it the most ever recorded. Phishing attacks are increasing despite of the use of e-mail filters. Although only 0.001 percent of these e-mails sent are answered, this percentage is enough to provide a return on investment and keep the phishing industry alive [24].

Nowadays, although different systems exist to deal with the problem of electronic fraud, these systems are far from optimal for classification purposes. The

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 321–328, 2007.
© Springer-Verlag Berlin Heidelberg 2007

methods underlying these systems can be classified into two categories, depending on the part of the message chosen by the filter as the focus for deciding whether e-mail messages are legitimate or fraudulent:

1. Origin-based filtering. Origin-based filters focus on the source of the e-mail and verify whether this source is on a white verification list [16], [25] or on a black verification list [15], [28].
2. Content-based filters focus on the subject and body of the e-mail. These can be also divided into two classes, depending on whether the analyzed content is textual or non-textual:
   (a) Textual content filters classify e-mails using Bayesian classifiers [18], [8], [26], heuristic rules [2], [19], case based reasoning [4], [3], [1], neural networks [27], support vector machines [5] or genetic algorithms [20].
   (b) Non-textual content filters examine the links embedded in e-mails by different techniques. Some filters check whether the links belong to white and black verification lists [7], [14], [6], [29]. Other filters analyze the appearance of the alphanumeric string of links and look for well known phishing schemes [22]. A third kind of filters [14] analyzes the longevity and hosting location of the websites addressed by links.
3. Other filters. Some of the filtering approaches adopted present a combination of techniques belonging to the previous categories. Of these, the most noteworthy is [10].

Most current commercial filters use verification lists to analyze e-mail senders and links contained in the body of e-mails. These filters are dynamic and update verification lists when new attacks are reported. However, the updating rate of filters is often overcome by the changing rate of the attacks because phishing e-mails are continuously modifying senders and link strings, many websites are only available one day, and continuous updating thus implies a high economic cost.

This paper describes a client-side system called FRALEC, which was designed and built to detect and filter phishing e-mail automatically, using different sources of information present in the content of e-mails which are handled by the processing methods most suitable for each information type. The system obtains optimum results and is highly effective at classifying e-mail, and highly efficient at managing resources.

2 The FRALEC System

The range of procedures used by a phisher, which evolve quickly in order to evade filters available, makes it necessary to solve the problem of identifying phishing e-mails with a multistrategy and integrated approach. The system shown in this paper focuses on a global view of all the information provided by an e-mail and the different analysis methods for this information.

One main feature of this system is effectiveness, measured by *precision* (percentage of predicted documents for a category that are correctly classified) and

recall (percentage of documents for a category that are correctly classified), because it applies three different classifiers in order to obtain all the information needed at each decision point to classify an e-mail serially. Another relevant feature is efficiency, because the order in which the classifiers are applied aims to obtain fast and reliable classifications by minimizing the resources used in a first classification step and then, if it is necessary, more complex methods are used in following classification steps. Figure 1 shows the flow that e-mails follow to be classified by the system.

Each classifier is trained on a predefined corpus. In the classification phase, the system decides to assign an e-mail to the *Fraud* or *Legitimate* categories after applying the classifiers described below in a serial way.

2.1 Bayesian Classifier

A Bayesian classifier [12] focuses on the textual content of e-mails to assign them to the *Economic* or *Non-Economic* categories. Since there are two parts in an e-mail, subject and body, which can contain text, a Bayesian classifier is built for each part. The lack of legitimate e-mails makes it difficult to train Bayesian classifiers that discriminate between the *Legitimate* and *Fraud* categories according to the textual content of e-mails. This is the reason for training the Bayesian classifier for each part with a corpus of e-mails labeled into the *Economic* and *Non-Economic* categories. In the training stage, the probabilities for each word conditioned to each category are estimated, and a vocabulary of words with their associated probabilities is created for both parts separately. Each filter classifies a new text into a category by estimating the probability of the text for each possible category C_j, defined as $P(C_j|text) = P(C_j)\Pi_i(word_i|C_j)$, where $word_i$ represents each word contained in the text to be classified. Once these computations have been performed, each Bayesian classifier assigns the text to the category that has the highest probability value.

The system integrates the classification decisions of both parts by assigning the e-mail to the *Economic* category whenever the classification result of any of the two parts is in the *Economic* category.

2.2 Rule Based Classifier

A rule-based classifier, which focuses on non-grammatical features of e-mails, classifies e-mails previously assigned to the *Economic* category by the Bayesian and assigns them to one of the *Legitimate*, *Fraud* or *Suspicious* categories. Since knowing whether an e-mail is economic is not sufficient to classify it into the *Legitimate* and *Fraud* categories, the goal of the system is only attained by analyzing other kinds of data contained in e-mails. The rule-based classifier uses the following rules, which have been obtained after applying a supervised learning algorithm [11] to the non textual content of the body part:

Rule 1: If the body of an economic e-mail contains forms, then the e-mail is assigned to the *Fraud* category.

Rule 2: If the body of an economic e-mail does not contain forms, images or links, then the e-mail is assigned to the *Legitimate* category.

Rule 3: If the body of an economic e-mail contains links in the text or links in an image, then the e-mail is assigned to the *Suspicious* category.

2.3 Emulator-Based Classifier

If the rule-base classifier is not able to classify an e-mail into the *Fraud* or *Legitimate* categories, then a third classifier emulates a fictitious access to the websites referenced by links contained in the body of the e-mails assigned to the *Suspicious* category. The emulation procedure consists of two steps: extracting the structure of the website addressed by the link, and filling in the website structure with fictitious data. Next, the emulator-based classifier analyzes the responses obtained from the fictitious access to these websites.

The classification model used for recognizing the response given by the website is based on the construction of a finite state automaton that represents words contained in a certain language and collects the grammar presented in this language. The language used in this case consists of the sentences contained in the responses given from legitimate financial websites to fictitious accesses. A grammar inference algorithm called ECGI [17] takes the sentences from the legitimate response base and the probabilities of all bigrams (groups of two words) and generates the automaton states and the edges between the states. Every edge connecting two states is weighted with the probability of the corresponding bigram. When the automaton receives a new sentence, the words of the sentence are processed according to their written order as in [21]. If there is a state representing the processed word and this state is reachable from the actual state, then the automaton follows that connection. If there is no state reachable representing the word, then the automaton follows the connection that has the highest probability. After following this connection, the automaton processes the next word. When the automaton reaches the final state, the automaton response is the sequence of words with the highest probability and similarity related to the sequence of words included in the sentence analyzed.

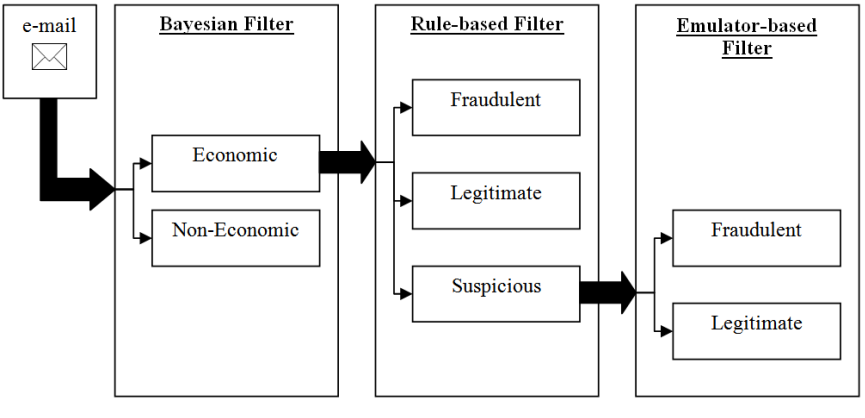


Fig. 1. Classification flow followed by an e-mail

The emulator-based classifier uses a function that allows computing the similarity between an input sentence and the sentence obtained by the automaton. The similarity function uses the number of overlapping words in both sentences and the document frequency of these words. If the similarity value computed is greater than an empirically determined threshold, then the classifier considers that the two sentences are similar. If this value is less than the threshold, then the sentences are not considered similar.

Therefore, if the emulator-based classifier considers that a textual response obtained after a fictitious access is similar enough to the automaton response, then the textual response is considered legitimate. So the e-mail that references the website is classified into the *Legitimate* category. If the responses are not similar enough then the e-mail is classified into the *Fraud* category.

3 Empirical Evaluation

The classifier system was evaluated on a collection of e-mails composed of 1,038 economic messages, divided into 10 legitimate messages and 1,028 fraudulent messages, and 1,006 non-economic messages. A small fraction (4 e-mails) of the legitimate economic messages and all non-economic messages were extracted from [23]. The remaining legitimate economic messages (6 e-mails) were messages received by the authors of this paper during a given period of time. It should be highlighted that it is very difficult to obtain financial legitimate e-mails because the financial sector is reluctant to use such means. This is primarily due to the growing number of phishing attacks. From 1,028 fraudulent economic messages, 833 were extracted from [13] and the remaining messages came from the inboxes of the authors of this paper. This collection was split into 5 folds in order to apply a 5-fold cross validation.

The textual content of the subject and body parts of messages was preprocessed by applying a stop list, and stemming the words. Next, words recurring below an experamentally determined threshold were removed.

Table 1 shows the *precision*, *recall* and *F-measure* (F = (2 * precision * recall) / (precision + recall)) values obtained by the integration of both textual Bayesian

Table 1. Performance measurements resulting from integrating the classifications obtained by the Bayesian classifiers on the subject and body parts of e-mails (Classes: *Non-Economic N-E, Economic E*)

	C1	C2	C3	C4	C5	Average
Precision N-E	0.956	0.976	0.995	0.995	0.926	0.969
Recall N-E	0.975	0.995	0.985	0.975	0.683	0.922
F-Measure N-E	0.965	0.985	0.989	0.984	0.786	0.942
Precision E	0.975	0.995	0.986	0.976	0.753	0.937
Recall E	0.957	0.976	0.995	0.995	0.937	0.972
F-Measure E	0.965	0.985	0.990	0.985	0.834	0.952

Table 2. Performance measurements of the rule-based classifier considering two classes (Classes: *Fraud F, Legitimate L*)

	C1	C2	C3	C4	C5	Average
Precision F	0.974	0.995	0.985	0.971	0.784	0.941
Recall F	0.950	0.933	0.981	0.990	0.894	0.949
F-Measure F	0.961	0.963	0.982	0.980	0.835	0.944
Precision L	0.953	0.934	0.980	0.990	0.862	0.943
Recall L	0.976	0.995	0.985	0.970	0.683	0.921
F-Measure L	0.964	0.963	0.982	0.979	0.762	0.930

Table 3. Performance measurements of the emulator-based classifier (Classes: *Fraud F, Legitimate L*)

	C1	C2	C3	C4	C5	Average
Precision F	0.954	0.989	0.971	0.971	0.925	0.962
Recall F	0.945	0.933	0.981	0.990	0.894	0.948
F-Measure F	0.949	0.960	0.975	0.980	0.909	0.955
Precision L	0.948	0.934	0.979	0.989	0.895	0.949
Recall L	0.957	0.990	0.970	0.970	0.926	0.962
F-Measure L	0.952	0.961	0.974	0.979	0.910	0.955

classifiers. The last column presents the macro averaged values and the other five columns present the result of each execution of the cross validation.

The decision rules of the rule-based classifier were learned from the fraudulent and legitimate economic e-mails of the first corpus by removing the textual content of the body part of e-mails. Each message is represented as a vector of three dimensions (links, images, forms). A supervised learning algorithm obtained the descriptions of both classes, *Fraud* and *Legitimate*. The overlapping descriptions of both classes were used to build the description of the *Suspicious* class. The effectiveness of this classifier in the *Fraud* and *Legitimate* categories is reported in Table 2 which is organized analogously to Table 1. In this experiment, the messages categorized into the *Suspicious* class were assigned to the *Fraud* class by default.

In order for the system to make a confident decision about the class of a message, messages assigned to the *Suspicious* class were finally processed by the third classifier. The emulator-based classifier accessed to the links contained in the messages and classified the textual response given by the website after accessing to it as a fictitious user. Table 3, which is organized as Table 1, shows the performance measurements of the emulator-based classifier when it classifies the messages previously assigned to the *Suspicious* class by the rule-based classifier. All effectiveness macro averaged values are better than those obtained by only applying the rule-based classifier.

4 Conclusions

FRALEC is an effective system for filtering fraudulent e-mails. Its good performance is reached because of integrating different classification methods that deal with all kinds of data present in e-mails. The procedure applied for evaluating the websites addressed by the links in e-mails allows the system to give high performance in terms of false negatives. FRALEC has been designed for client-side operation. However, thanks to its underlying inner nature, it does allow for straightforward expansion to multiple-user support.

References

1. Androutsopoulos, I., Paliouras, G., Karkaletsis, G., Sakkis, G., Spyropoulos, C., Stamatopoulos, P.: Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. In: Workshop on Machine Learning and Textual Information Access. 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (2000)
2. Cohen, W.: Learning rules that classify e-mail. In: AAAI Spring Symposium on Machine Learning in Information Access (1996)
3. Cunningham, P., Nowlan, N., Delany, S.J., Haahr, M.: A Case-Based Approach to Spam Filtering that Can Track Concept Drift. Technical Report at Trinity College, TCD-CS-2003-16, Dublin (2003)
4. Daelemans, W., Zavrel, J., van der Sloot, K., van den Bosch, A.: TiMBL: Tilburg Memory-Based Learner - version 4.0 Reference Guide (2001)
5. Drucker, H., Wu, D., Vapnik, V.N.: Support Vector Machines for Spam Categorization. IEEE Transactions on Neural Networks 10(5) (1999)
6. GeoTrust TrustWatch, `http://www.trustwatch.com/`
7. GoDaddy, `http://www.godaddy.com/`
8. Graham, P.: Better Bayesian Filtering. In: Proc. of Spam Conference 2003, MIT Media Lab, Cambridge (2003)
9. June Phishing Activity Trends Report (2006), `http://www.antiphishing.org`
10. McAffee SpamKiller, `http://www.spamkiller.com`
11. Michalsky, R.S.: A theory and methodology of inductive learning. In: Michalsky, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) Machine Learning: An Artificial Intelligence Approach, pp. 83–134. Springer, Heidelberg (1983)
12. Mitchell, T.M.: Machine Learning. McGraw-Hill, New York (1997)
13. Monkey.Org Inc., `http://www.monkey.org/~jose/wiki/doku.php`
14. Netcraft, `http://news.netcraft.com/`
15. Pyzor, `http://pyzor.sourceforge.net`
16. Randazzese, V.A.: ChoiceMail Eases Antispam Software Use While Effectively Fighting Off Unwanted E-mail Traffic. CRN (2004)
17. Rulot, H.: ECGI. Un algoritmo de Inferencia Gramatical mediante Corrección de Errores. Phd Thesis, Facultad de Ciencias Físicas, Universidad de Valencia (1992).
18. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)
19. Sergeant, M.: Internet-Level Spam Detection and SpamAssassin 2.50. In: Proceedings of Spam Conference 2003, MIT Media Lab. Cambridge (2003), `http://spamassassin.org`

20. Serrano, J.I., Castillo, M.D.: Evolutionary Learning of Document Categories. Journal of Information Retrieval 10, 69–83 (2007)
21. Serrano, J.I., Araujo, L.: Statistical Recognition of Noun Phrases in Unrestricted Text. In: Famili, A.F., Kok, J.N., Peña, J.M., Siebes, A., Feelders, A. (eds.) IDA 2005. LNCS, vol. 3646, pp. 397–408. Springer, Heidelberg (2005)
22. Spoofstick, http://www.spoofstick.com/
23. SpamAssassin, http://spamassassin.apache.org/publiccorpus/
24. Suckers for spam (2005), http://www.Internetnews.com
25. Tagged Message Delivery Agent Homepage, http://tmda.net
26. Understanding Phishing and Pharming. White Paper (2006), http://www.mcafee.com/
27. Vinther, M.: Junk Detection using neural networks. MeeSoft Technical Report (2002), http://logicnet.dk/reports/JunkDetection/JunkDetection.htm
28. Razor, V.: http://razor.sourceforge.net
29. 3Sharp LLC, Gone Phishing: Evaluating Anti-Phishing Tools for Windows, Robichaux P., Ganger, D.L., http://3sharp.com

Formal Reasoning on a Web Coordination System*

Eloy J. Mata[1], Pedro Álvarez[2], José A. Bañares[2], and Julio Rubio[1]

[1] Departamento de Matemáticas y Computación, Universidad de La Rioja, Edificio Vives, Luis de Ulloa s/n, E-26004 Logroño (La Rioja, Spain)
{eloy.mata,julio.rubio}@unirioja.es
[2] Departamento de Informática e Ingeniería de Sistemas, Universidad de Zaragoza, María de Luna 3, E-50015 Zaragoza, Spain
{alvaper,banares}@unizar.es

Abstract. In this paper, a first step toward the use of Artificial Intelligence tools (namely proof assistants) in the formal analysis of programs for Web services coordination is presented. This first attempt consists in the formal modeling of a system with transactional capabilities. The model is devised on a variant of the well-known Linda model for generative communication. We explore then the role of the Rete algorithm to implement efficiently a transactional `read` operation, opening the way for a further formal analysis of it, by means of automated testing against a certified program (i.e. a program verified with the help of a proof assistant).

1 Introduction

In the paper [1] the role of mechanized reasoning tools (as ACL2, Isabelle or Coq) in the verification of software systems, specifically in the field of algebraic manipulation, has been studied.

In an apparently far domain, Web services coordination, some proposals have been made (see for instance [8]) to enhance and implement coordination systems based on the Linda model [2] (by means of software frameworks such as JavaSpaces or TSpaces).

This paper stems from the confluence of these two areas. The theoretical tools used to model computer algebra programs are different from those employed in Linda-based Web coordination systems. In the first case, algebraic specifications are used [1]; in the second one, the formal methods are based on process algebras and transition systems (see [3], for instance). Nevertheless, our objectives in both cases are similar: to model formally already-written programs and, in a second step, to increase their reliability by means of proof assistants.

In previous works ([4,5]), the task of modeling the JavaSpaces-based Web Coordination Service (WCS) described in [8] was undertaken. In this paper, we

* Partially supported by Comunidad Autónoma de La Rioja, project ANGI2005/19, and Ministerio de Educación y Ciencia, projects MTM2006-06513, TIN2006-13301.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 329–336, 2007.
© Springer-Verlag Berlin Heidelberg 2007

go beyond the "atomic" Linda model, enhancing it with transactional properties, which are considered necessary to complex coordination tasks. Our main results consists in extending with *constraints* the formal Linda model, enabling more complex queries, and then giving over it a multi-tuple selection mechanism. A good strategy to implement efficiently this formal model is by adapting the well-known Rete algorithm [6,7], providing the extended Linda semantics is maintained. We will explain how this reliable strategy fits into our modeling setting.

This paper is organized as follows. In Section 2 generalities, on both Web services coordination and Linda-based models, are introduced. The formal treatment appears in Section 3, and then, in Section 4 the role of the Rete algorithm is explained. Finally some conclusions and open problems are presented.

2 Web Coordination and Linda-Based Models

The WCS reported in [8] acts as a message broker, that is, it supports message-based interoperability among Web services. Web services can send messages to, and receive from, the WCS which allows an asynchronous communication model and a loosely couple architecture for distributed applications by exchanging messages through this single place.

WCS is conceptually inspired by Linda. Linda is a well-known proposal, based on Generative Communication [2] and is composed of a small set of operators, acting on a blackboard, called *tuple space*, together with a pattern-matching oriented process to deal with tuples.

A Linda tuple is something similar to `["vehicle1",41323,0.8323]`, a list of untyped atomic values. We had to enrich some structural aspects of Linda. Our main extension implies passing from this atomic organization to lists of attribute/value pairs, as in: `[<id,"vehicle1">,<lat,41323>,<long,0.8323>]`. Although this is still an untyped setting, this bit of structure allows information recovery in distributed environments. The reason is that this kind of structure corresponds with some simple types of XML documents (the tags playing the role of attributes). In addition, the pattern-matching process associated to Linda can be now considered as a complex procedure in which attributes act as keys for the matching.

These features have been experimentally found when developing our WCS: they are needed to deal with XML documents encapsulated as Java objects (let us stress that JavaSpaces is the basic technology for our Service). Thus, this complex matching works in two phases. First, the matching is guided by attributes in order to select a set of objects representing tuples with the same XML structure. This set of objects is called a *channel*, and it is an important notion both from the operational and from the formalization points of view. In the second phase, the matching is done by values on the channel, to obtain a valid tuple. The designed WCS is composed of three software components(see Figure 1): (i) the *XML-based Space* component which encapsulates the tuple space and provides a collection of operations to write XML-tuples into and to

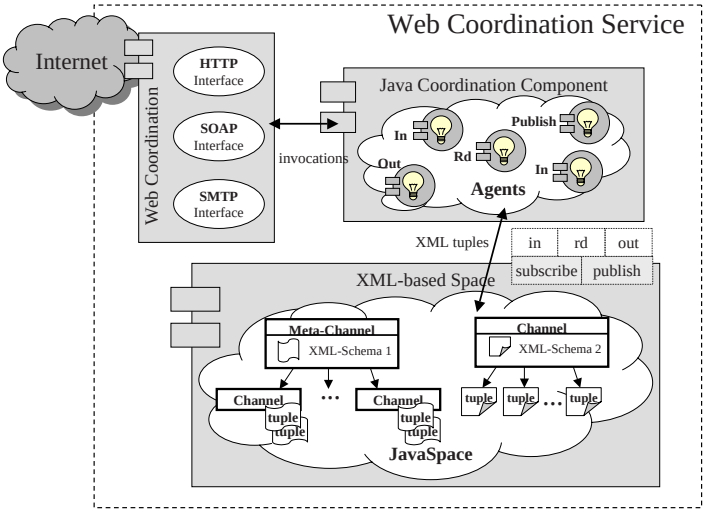


Fig. 1. Software Components of the coordination service

read or remove them from the tuple space; (ii) the *Java Coordination* component; and (iii) the *Web Coordination* component (for the details see [8]).

This Service has been used in a real project (oriented to location-based Web services [8]) to show its practical value. Nevertheless, the clear semantic concepts of Linda have been shadowed by several layers of complex wrappings from XML documents to Java objects. So, a main concern is whether the final operations for reading and writing complex tuples really behave as Linda operators. To increase the reliability of our WCS, our strategy implied giving a formal model of our blackboard architecture, and then proving its correctness. This was done in [5], and is briefly explained at the beginning of the next section.

3 A Formal Model with Transactional Capabilities

In [3] the tuple-based coordination medium is represented as a software component interacting with coordinated entities by receiving *input events* and sending *output events*. The main elements of this model are: a set of tuples t ranging over T ($\overline{T}$ denotes the set of multisets over T); a set of templates $templ$ ranging over $Templ$; a matching predicate $mtc(templ, t)$ between templates and tuples; and a choice predicate $\mu(templ, \overline{t}, \widehat{t})$, where $\overline{t}$ is a multiset of tuples ranging over $\overline{T}$, and the symbol $\widehat{t}$ means that it can be a tuple t from the tuple multiset $\overline{t}$ which matches the template $templ$, or can be an error element $\perp_T$ if no matching is available in $\overline{t}$. This intended behavior is formally defined in the following matching rules, where $t|\overline{t}$ is used to denote the union of the element t and the multiset $\overline{t}$ (see [3] for details):

$$\frac{mtc(templ,t)}{\mu(templ,t|\overline{t},t)} \qquad \frac{\nexists\ t\in\overline{t}\quad mtc(templ,t)}{\mu(templ,\overline{t},\perp_T)} \tag{1}$$

In order to provide a formal model closer to our WCS, in [5] we worked with a version of Linda where tuples and templates are sequences of attribute/value pairs; we called this Linda version *structured model*. In this model, a *structured-tuple* is a finite and ordered sequence of attribute/value pairs, and has the form $t_{str} = ((a_1, v_1)\dots(a_n, v_n))$ and a *structured-template* is a finite and ordered sequence of attribute/value pairs where values can be either data or wildcard variables with the form $templ_{str} = ((ta_1, tv_1)\dots(ta_m, tv_m))$. The structured-matching predicate $mtc_{str}(templ_{str}, t_{str})$ holds if both t_{str} and $templ_{str}$ have the same attribute structure and, each non-wildcard field of $templ_{str}$ is equal to the corresponding field of t_{str}. The soundness of this new model (i.e., the proof that the Linda equations hold on this new model) has been proved in [5].

One of the main limitations of the basic Linda model is that the basic coordination primitives involve only a single tuple, while the realization of transactions needs several operations acting on more than one tuple [9]. One approach for the realization of transactions is to ensure the preservation of the so called ACID properties. Another alternative approach is to extend the set of Linda primitives by adding operations to extract or insert atomically multisets of data. Some examples are the `multwrite` primitive in TSpaces; the `rew` operation modeled in [9]; the `collect` primitive in [10]; or the `min` operator discussed in [11].

We have adopted this second approach extending the structured model with a multiple template matching predicate to extract atomically a set of tuples instead of a single item. Although some bulk primitives could be simulated by sequential calls to the basic Linda primitives, it has been proved (see [11]) that it is not possible to provide a general encoding of a multiple reading operation using the standard Linda operators, because a sequential reading of the tuples does not guarantee the semantics of the multiple reading operation.

Based on the structured model, we need now an extended syntax, richer than the simple constant concordance, to describe restrictions in templates which allow us the use of a query language in order to obtain tuples that match with respect to more complex criteria. In this enhanced model, called *model with constraints*, a template is a finite sequence of attribute/constraint pairs as $templ_{res} = ((a_1, r_1)\dots(a_m, r_m))$, where each r_i is a coherent and well-formed constraint according to the BNF notation defined in Table 1. Now, a tuple matches a template if every constraint included in the template is satisfied by the tuple, and a constraint is satisfied by a tuple if the constraint returns *true* when it is evaluated after all the variables are binded by the corresponding values of the tuples (let us stress that in a restriction r_i variables related to different templates can occur, allowing inter-tuple constraints to be expressed).

From now on and in the rest of this Section, in order to ease the reading, we will work in the simplest case of two templates (the generalization to the n-tuples case is straightforward). The *multiple-matching predicate* now has the form (in the case of two templates):

$$mtc2(templ_1, templ_2, t_1, t_2)$$

Table 1. BNF form for constrained templates

$$
\begin{array}{lll}
\langle template \rangle & ::= & ((\langle attribute \rangle, \langle constraint \rangle)^+) \\
\langle constraint \rangle & ::= & ? \quad | \\
 & & (\langle constraint \rangle) \quad \& \quad (\langle constraint \rangle) \quad | \\
 & & (\langle constraint \rangle) \quad || \quad (\langle constraint \rangle) \quad | \\
 & & \langle literal \rangle \\
\langle literal \rangle & ::= & \langle atom \rangle \quad | \\
 & & \neg \langle atom \rangle \\
\langle atom \rangle & ::= & (\langle predicate \rangle \quad \langle term \rangle^+) \\
\langle term \rangle & ::= & \langle constant \rangle \quad | \\
 & & ? \langle variable \rangle \quad | \\
 & & (\langle function \rangle \quad \langle term \rangle^+) \\
\langle predicate \rangle & ::= & < \quad | \quad > \quad | \quad \leq \quad | \quad \geq \quad | \quad = \quad | \quad \neq \\
\langle function \rangle & ::= & + \quad | \quad - \quad | \quad * \quad | \quad /
\end{array}
$$

where $mtc2$ predicate is satisfied if the following conditions hold:

1. The length of $templ_1$ is equal to the length of t_1 and both of them are defined over the same set of attributes.
2. The length of $templ_2$ is equal to the length of t_2 and both of them are defined over the same set of attributes.
3. Each constraint in both templates $templ_1$ and $templ_2$ is well defined over the total set of attributes.
4. Each constraint in both templates $templ_1$ y $templ_2$ is satisfied by the values of both tuples t_1 and t_2.

Given the multiple-matching predicate $mtc2$, a multiple-choice predicate $\mu 2$ is also defined satisfying Linda matching rules of (1). Now we are able to define new operations involving multiple tuples. For example, operations such as $out2()$, $rd2()$, $in2()$ or $rd.in()$ can write, read or remove two tuples atomically. In particular, the formal specification of the multiple reading operation $rd2()$ is:

$$\frac{mtc2(templ_1, templ_2, t_1, t_2)}{S(rd2(templ_1, templ_2)^o, t_1|t_2|\overline{t})}$$

$$\frac{mtc2(templ_1, templ_2, t_1, t_2)}{\langle t_1|t_2|\overline{t}, \overline{w}, \underline{o}t_1|\underline{o}t_2, t_1|t_2|\overline{t}, \overline{w}\rangle \in E(rd2(templ_1, templ_2)^o)}$$

In other words, given two templates $templ_1$ and $templ_2$, the operation $rd2(templ_1, templ_2)$ atomically returns two tuples t_1 and t_2 from the tuple space, complying with the four conditions previously enumerated, when such tuples exist, or the invoking process is blocked until both tuples appear in the tuple space. In the sequel, this structure composed of returned tuples as t_1 and t_2 (perhaps still incomplete) will be called, according to the Rete algorithm terminology, a *token*.

4 Formal Analysis of a Rete Algorithm

In this Section we will give an overview of the role of the Rete algorithm in our transactional coordination service. An informal account was done in [4], where an example of application can be found. In the rest of the Section we will work with the n-tuples version of our transactional model.

We are looking for an efficient mechanism, and reliable in a distributed setting, which implements the multiple reading operation $rd_n()$ over a tuple space TS. Some similarities between rule-based languages and Linda-based models (as facts vs. tuples, rules vs. templates, or assert/retract operations vs. out/in primitives) led us toward an implementation of a variant of the Rete algorithm. Although the use of a Rete-like algorithm suggests the possibility of obtaining an efficient program, some difficulties arise when we attempt its formal analysis, due to the logical complexity of the Rete algorithm, which acts as a "black box" in many popular expert systems shells, including OPS5, CLIPS, JESS and Soar. In this case, even testing could be difficult because of its deployment in a distributed context.

To overcome these difficulties, our aim is to follow the same ideas explained in [1]. First, trying to verify a simplified (but inefficient) version of the algorithm; and then in a second step, testing the efficient version, which is supposedly difficult to verify, against the inefficient (but already verified) version.

The Rete network structure may improve the efficiency in two aspects: it stores partial matches when performing joins between different templates, so Rete avoids re-computing from scratch after each modification (writing or removing tuples) of the tuple space; and, in addition, it takes advantage of the similarity between templates, in the same multiple reading operation (or even in different reading operations; our current approach does not consider this possibility), in such a way that when two or more templates have a common constraint they share the same node, thus reducing certain types of redundancy.

Function `rete_based_multiple_rd`(*lot:list of templates*):list of tuples

```
r ← create_rete_network(lot);
propagate_tuples_from_TS_to_rete(r);
while not query_completed_matched?(r) do
    op ← listen_to_a_Linda_operation ;
    switch op do
        case writing_tuples add_wme(tuples,r);
        case removing_tuples remove_wme(tuples,r);
    end
end
return token
```

In this function the first block is devoted to constructing an initial Rete network by means of the operations `create_rete_network` and `propagate_tuples_from_TS_to_rete`. The predicate `query_completed_matched?` checks if there is a *token* in the network given a complete solution to the $rd_n()$ query (if this is the case, it is assumed that the

meta-variable *token* is suitably updated). This first block must be executed atomically (with respect to the state of the tuple space TS). If no solution is reached, the process comes into a loop, where the procedures `add_wme` and `remove_wme` modify memory nodes in the network (here we follow the terminology and ideas of [7], where the Rete algorithm is specified in detail).

Instead of trying to analyse the previous algorithm formally, our idea is to devise a simpler procedure, behaviorally equivalent to the first one. Let us remark, first, that the first block of the Rete-based algorithm should be behaviorally equivalent to the following simple process. It uses the notion of channel and a database-like strategy to determine if, in a given snapshot of the tuple space TS, there is a list of tuples satisfying the input templates. The output of the process is a *token*, given an answer (perhaps incomplete) to the query.

Function `snapshot_multiple_rd`(*lot:list of templates*):list of tuples

```
t ← ();
foreach template in lot do
    c ← obtain_the_corresponding_channel_from_TS(template);
    t' ← select_tuples_satisfying_local_tests(c);
    t ← append(t,t');
end
token ← join_selected_tuples(t);
return token
```

In this function, a list of tuples t is first binded to the empty list. Then for each input template, the corresponding channel c is obtained from the tuple space TS. The operation `select_tuples_satisfying_local_tests` gets the tuples of c which satisfy the local constraints (that is to say, the constraints involving just one tuple of the channel). The resulting tuples t' are sequentially concatenated. When this preprocessing is finished, the operation `join_selected_tuples` performs the appropriate variable bindings consistency checks on the list of tuples t, with respect to the constraints relating several tuples.

If, as intended, this process is equivalent to the first block in the Rete-based algorithm, then the correctness of the following multiple reading algorithm should be easily verified.

Function `inefficient_multiple_rd`(*lot:list of templates*):list of tuples

```
token ← snapshot_multiple_rd(lot);
while not query_completed_matched?(token) do
    listen_to_a_Linda_operation ;
    if TS is changed then
        token ← snapshot_multiple_rd(lot);
    end
end
return token
```

Once the properties of this (extremely) inefficient algorithm were proved with the help of a proof assistant, the automated testing of the Rete-based algorithm could be undertaken, thus increasing its reliability.

5 Conclusions and Further Work

In this paper we have reported on a project to apply formal methods in applied Software Engineering, and more concretely in the field of coordination of Web Services. We have explained how proof assistants could play a role in this area of distributed computing, by aiding to prove properties of simple coordination programs. These inefficient programs would be used, later on, to do automated testing of other more efficient ones, such as that based on the Rete algorithm to achieve multiple-tuple reading in a transactional coordination model.

According to the tentative nature of our proposals, many research lines are still open, ranging from the experimental study of the performance of the Rete algorithm in this context, to the beginning of the use of mechanized reasoning tools to verify properties of simple auxiliary programs.

References

1. Andrés, M., Lambán, L., Rubio, J.: Executing in Common Lisp, Proving in ACL2. In: Calculemus 2007. LNCS, vol. 4573, pp. 1–12. Springer, Heidelberg (2007)
2. Gelernter, D.: Generative communication in Linda. ACM Transactions on Programming Languages and Systems 7(1), 80–112 (1985)
3. Viroli, M., Ricci, A.: Tuple-Based Coordination Models in Event-Based Scenarios. In: IEEE 22nd International Conference on Distributed Computing Systems (ICDCS 2002 Workshops) - DEBS'02 International Workshop on Distributed Event-Based Systems, Vienna, Austria (July 2002)
4. Mata, E., Álvarez, P., Bañares, J.A., Rubio, J.: Towards and Efficient Rule-Based Coordination of Web Services. In: Lemaître, C., Reyes, C.A., González, J.A. (eds.) IBERAMIA 2004. LNCS (LNAI), vol. 3315, pp. 73–82. Springer, Heidelberg (2004)
5. Mata, E., Álvarez, P., Bañares, J.A., Rubio, J.: Formal Modelling of a Coordination System: from Practice to Theory and back again. In: Mata, E., Álvarez, P. (eds.) Engineering Societies in the Agents World VII, ESAW 2006, Dublin, Ireland. LNCS (LNAI), Springer, Heidelberg (2007)
6. Forgy, C.: A fast algorithm for many pattern / many object pattern match problem. Artificial Intelligence 19, 17–37 (1982)
7. Doorenbos, R.B.: Production Matching for Large Learning Systems (Rete/UL). PhD thesis, Carnegie Mellon University (1995)
8. Álvarez, P., Bañares, J.A., Muro-Medrano, P.: An Architectural Pattern to Extend the Interaction Model between Web-Services: The Location-Based Service Context. In: Orlowska, M.E., Weerawarana, S., Papazoglou, M.M.P., Yang, J. (eds.) ICSOC 2003. LNCS, vol. 2910, pp. 271–286. Springer, Heidelberg (2003)
9. Busi, N., Ciancarini, P., Gorrieri, R., Zavattaro, G.: Coordination Models: A Guided Tour. In: Coordination of Internet Agents, pp. 6–24. Springer, Heidelberg (2001)
10. Rowstron, A., Wood, A.: Solving the Linda multiple `rd` problem using the `copy-collect` primitive. Science of Computer Programming 31(2-3), 335–358 (1998)
11. Zavattaro, G.: Coordination Models and Languages: Semantics and expressiveness. PhD thesis, Department of Computer Science, University of Bologna, Italy (2000)

A Multi-agent System for Information Retrieval

Lenka Lhotska[1] and Laura Prieto[2]

[1]Gerstner Laboratory, Czech Technical University in Prague
Technicka 2, 166 27 Prague 6, Czech Republic
lhotska@fel.cvut.cz
[2]Universitat Rovira i Virgili
Tarragona, Spain
lpr_salamanca@hotmail.com

Abstract. This paper is focused on description of a partial project of a multi-agent system for information retrieval. It is an integral part of a large ongoing project: the design of a multi-agent system for supporting medical diagnostics and monitoring in the area of cardiology (the ADAM system). Cooperating agents provide a very natural means of automating the pre-processing and (at least partially) the evaluation of a vast amount of medical data utilizing all available medical knowledge. As additional and very important functionality we have decided to design and implement agents that serve for building ontologies from medical resources on the Internet. The current phase of the project is targeted at implementing the system and putting it into full use.

Keywords: multi-agent system, information retrieval, ontology, medical information.

1 Introduction

The paper discusses recent results of the development of a multi-agent system for information retrieval in medical domain. Medicine is characterized by distributed data, information, knowledge, and competence. Large volume of information is available in Internet resources (e.g. PubMed). However it is impossible to perform search manually using for example Google. It is more efficient to develop tools that allow users to find relevant information in a more suitable way and use it for the decision support. Distributed problem solving and agent technology offer efficient and natural solutions, because they correspond to the main properties of the medical domain. Recently there has been growing interest in the application of agent-based systems in health care in general. The most frequent medical domains in which agents have already been considered are: retrieval of medical knowledge from the Internet [1], decision support systems for monitoring and diagnostic tasks [2] or for home care, and distributed patient scheduling within a hospital [3].

2 Related Work

Information retrieval represents a challenging task because it usually deals with natural language which is not always well structured and may be semantically

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 337 – 344, 2007.
© Springer-Verlag Berlin Heidelberg 2007

ambiguous [4]. Recent development in information retrieval from distributed data and knowledge resources shows that intelligent, more sophisticated tools are necessary in order to make the work more efficient and user friendly. We will briefly describe several examples of systems using agent technology for information retrieval.

In [5] the authors describe an experimental mobile-agent system searching technical reports distributed across multiple machines. The application was implemented on top of the D'Agents system [6]. It utilizes the statistical information-retrieval system Smart that uses the vector-space model to measure the textual similarity between documents and is wrapped inside a stationary agent (Smart IR agent) on each machine. The main application agent is a GUI that runs on the user's machine. Menczer [7] designed and implemented Myspiders, a multi-agent system for information discovery in the Internet and showed that augmenting search engines with adaptive intelligent search agents can lead to significant competitive advantages. Nguyen with his colleagues [8] developed an agent system that assists at information retrieval from the Internet. It uses multiple agents for the retrieval tasks and consensus methods for resolving differences in response sets. There is one manager agent and several search agents that are created by the manager agent to utilize multiple search engines. The multiagent system developed by Mc Dermott and O'Rierdan [9] is applied to the domain of information management. The agents have three main components in their architecture: knowledge, beliefs and capabilities. Each agent has the ability to search its own document collection. In [10] the MELISA - Medical Literature Search Agent – a prototype of an ontology-based information retrieval agent is described. The main parts and functions are: query generation and reformulation, query evaluation, filter and combination, query model, medical ontology, MeSH browser. It works with MedLine and its associated web-based search engine – PubMed. In medical ontology it contains some medical knowledge used to generate the queries. JITIR agents (just-in-time information retrieval agents) [11] proactively present information based on a person's context. The information can come from any number of preindexed databases of documents. A key feature of a JITIR agent is that it can provide information from an unordered set of documents. No hand-coding of documents or special domain-dependent techniques are required.

3 Motivation

We focused on the possibility of utilizing a multi-agent system as a platform for development of an intelligent system for medical diagnostics and monitoring [12]. During implementation it proved to be useful to include the info-collecting agent. This agent should find relevant information in the Internet resources, extract and analyze it, and finally build ontologies from the retrieved information. During the process of design and implementation it has turned out that a single agent would be too complicated. Therefore the final decision has been to develop a small multi-agent system with required functionality (info-collecting system) that will communicate with the ADAM system [12]. Since the final aim is a decision support system in medicine we did not want to search for information in the whole Internet but to use reliable and relevant source of medical information. Therefore we decided to use MedLine with its search engine PubMed.

4 Tools Used in Implementation

We were looking for such an environment that would maintain basic multi-agent necessities, which are communication standards, agent administration etc. FIPA (Foundation for Intelligent Physical Agents) [13] provides a set of such standards and reference implementation that have emerged from industrial needs and achievements of the research community. Since the ADAM system is being built using JADE software framework [14], we also used JADE for this partial project. Further we tried to identify suitable tools for information extraction and building ontologies that would satisfy the basic requirements, namely being free or open source, and being easy to use. These tools are briefly described below.

JADE is a middle-ware (written entirely in the Java language, using several Java technologies), which simplifies the implementation of multi-agent systems by providing a set of graphical tools that support the debugging and deployment phases. In JADE, the agent platform can be distributed across machines (which not even need to share the same operating system) and the configuration can be controlled via a remote graphical user interface (GUI). The configuration can be even changed at run-time by moving agents from one machine to another one, when required. The platform independency is important advantage of JAVA solutions. JADE has several interesting features that at least make the process of implementation easier. One of these features is the agent Sniffer that enables user to observe message flow among agents. Another one is the existence of the Agent Management System (AMS). AMS is a mandatory component of the agent platform (AP) and only one AMS exists in a single AP. The AMS is responsible for managing the operation of an AP, such as the creation of agents, the deletion of agents, deciding whether an agent can dynamically register with the AP and overseeing the migration of agents to and from the AP.

PubMed [15], available via the NCBI Entrez retrieval system, was developed by the National Center for Biotechnology Information (NCBI) at the National Library of Medicine (NLM), located at the National Institutes of Health (NIH).

Entrez [16] is the text-based search and retrieval system used at NCBI for services including PubMed, Nucleotide and Protein Sequences, Protein Structures, Complete Genomes, Taxonomy, OMIM, and many others. PubMed was designed to provide access to citations from biomedical literature. LinkOut provides access to full-text articles at journal Web sites and other related Web resources.

HTMLParser [17] is a super-fast real-time parser for real-world HTML. What attracted most developers to HTMLParser was its simplicity in design, speed and ability to handle streaming real-world html. The two fundamental use-cases that are handled by the parser are extraction and transformation. It contains filters, visitors, own tags and JavaBeans easy to use. It is a fast, robust and well-tested package.

Lucene [18] from the Apache Jakarta project is a full-featured text search engine library written entirely in Java. It is a high-performance indexing and search engine. “Indexing” means to split sentences into the individual words for storing them in some kind of directory. This directory can then be used for fast lookup of words or combination of words.

Snowball [19] is a language in which stemming algorithms can be exactly defined, and from which fast stemmer programs in ANSI C or Java can be generated. A range

of stemmers is presented in parallel algorithmic and Snowball form, including the original Porter stemmer for English.

WordNet [20] is a lexical database for the English language. It groups English nouns, verbs, adjectives and adverbs into sets of synonyms called synsets; it provides short, general definitions, and records various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications.

OWL [21] is an acronym for **Web Ontology Language**, a markup language for publishing and sharing data using ontologies on the Internet. OWL is a vocabulary extension of the Resource Description Framework (RDF) and is derived from the DAML+OIL Web Ontology Language.

Protégé [22] is a free, open-source ontology editor and knowledge-base framework. It is a platform that provides users with a suite of tools to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. Further, Protégé can be extended by a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications.

5 Multi-agent System for Information Retrieval and Ontology Building

The designed multi-agent system has two different kinds of agents: Builder agent and Searcher agent (see Figure 1). During the process only one builder agent is created. The number of searcher agents depends on the parameters entered by the user, to be exact, on the number of required searches. So, initially only the builder agent is created, and it has to generate the searcher agents dynamically.

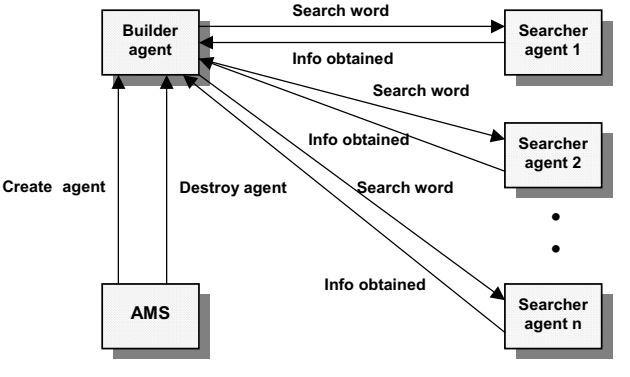


Fig. 1. Architecture of the information-collecting system

The user can enter several parameters using the graphic interface of the system. These parameters are necessary to build the ontology (see Table 1). The user must enter, at least, the main concept from which the ontology will be built. The other parameters are optional; they already have a default value assigned. The ontology is built in OWL language from these parameters and it is stored in a file, so that it can be read later on with a tool for reading ontologies like Protégé. Moreover, the ontology and the articles related to each class are displayed using a graphic user interface at the same time as they have been generated by the system, and the user can visualize them any time. The ontology is displayed in tree form and the user can see the title of the articles related to one of the classes by selecting it in the tree. In addition, if the user clicks on one of the articles, its reference in PubMed will be opened in an Internet browser.

5.1 Tasks of Builder Agent and Search Agents

During the process only one builder agent is created. The number of searcher agents depends on the parameters entered by the user, to be exact, on the number of required searches. So, initially only the builder agent is created, and it has to generate the searcher agents dynamically.

The tasks of the builder agent are: Load the graphical user interface to request the parameters to the user and display the result generated in the process (the hierarchy of classes and the articles related to each class). Generate the required searcher agents and destroy them when they have finished their work. Communicate with each searcher agent to notify the task it has to perform and get the result of this action. Analyse the information sent by the searcher agents and discard part of it if necessary. Build the ontology at the same time as it receives the required information by the searcher agents until the depth level set by the user is reached or until the process cannot continue because there is not information to create new subclasses.

The tasks of each searcher agent are: Search in PubMed the articles related to the class that the builder agent has assigned to it. Analyse the articles obtained and choose the subclasses that have to be generated from that class. Send the information to the builder agent (the new subclasses of the ontology and the articles related to each of them).

5.2 System Operation

When the user enters the main concept from which the ontology will be built (e.g. heart), the builder agent generates the first searcher agent so that it performs the search of this first class. Example of the input parameters is shown in Table I.

The searcher agent searches in PubMed the class name using the ESearch tool from the Entrez Programming Utilities. This tool returns a web page with a set of Ids, which correspond to the articles related to the class name. The agent parses this web page to extract these IDs using the HTMLParser tool and searches in PubMed the corresponding articles using the EFetch tool from the Entrez Programming Utilities. This tool returns a web page with the information of the articles and the agent parses it to extract the title and summary of each one using the HTMLParser tool.

Table 1. Example of Input Parameters

Item	Value
Main concept	**Heart**
Highest depth level	**3**
Highest number of articles to analyze for each class	**200**
Highest number of subclasses to generate for each class	**5**
Percentage or number of articles	**Percentage**
Lowest percentage of articles to generate a new class	**0**
Lowest number of articles to generate a new class	**----------**

For each article summary, the searcher agent analyzes the text separating it by words using the StopAnalyzer of the Lucene tool, which takes charge of ignoring everything what are not letters (e.g. symbols and numbers) and the stop words. After that, the searcher agent makes several checks with each word of the text, and when a word does not satisfy one of the constraints, it is rejected with no need to continue making the rest of the checks with it. When the checks are finished the searcher agent has to choose, among the words that have satisfied all the constraints, the *n* that have appeared in the articles a greater number of times, being *n* a parameter entered by the user, *n* is usually in the range (5, 15). Moreover, this parameter is the highest number of subclasses that each class will have. But there is still the last constraint that these *n* words have to satisfy: they must have appeared in a minimum number of articles or in a minimum percentage of the number of analyzed articles, according to the criterion that the user has chosen.

Once the searcher agent has the definitive words (e.g. breast and prostate) and the new subclasses are created in the ontology (e.g. breast cancer and prostate cancer), it sends the following information to the builder agent:

- The new subclasses that it has to add to the ontology.
- The title and ID of the articles obtained searching the parent class, but only those that no contains the new subclasses. For example, if it has searched the class ‘cancer’ and it has obtained the subclasses ‘breast cancer’ and ‘prostate cancer’, it will send as articles related to ‘cancer’ all those that it has analyzed where appears ‘cancer’ but not ‘breast cancer’ nor ‘prostate cancer’.

When the builder agent receives the information from the searcher agent, destroys it and analyzes one by one the new classes that it has received, discarding those that are equivalent to any class that it received previously, for any of the following reasons: because they contain the same words but in different order; because they are distinguished only by words from the same family (with the same root) (this check is made using the Snowball tool); because they are distinguished only by words that are synonymous, in any of its different meaning as noun or in any of its different meaning as adjective (this check is made with the WordNet tool).

After making the checks, the builder agent adds the definitive subclasses, the relationship between each of them and their parent class, and the articles related to

each one to the ontology. Then, the builder agent generates so many searcher agents as new classes have been created, and it assigns one of these classes to each one (in our example, two searcher agents are generated, one that will take charge of searching 'breast cancer' and the other will take charge of searching 'prostate cancer'). In this way, we return to the point in which each searcher agent searches its assigned class in PubMed, gets the text of the articles, analyzes it, etc. The process finishes when we arrive at the level of depth fixed by the user or at a level in which we are not able to make a new decomposition.

5.3 Agent Communication

The protocol used for the communication (transmission of messages) between the builder agent and the searcher agents, is the FIPA-Request protocol. It fits best with the needs of the process, since the builder agent has to send a message to each searcher agent with the action that it must perform and the required parameters for that, and later the builder agent has to receive the result of this action. The builder agent initiates a FIPA-Request protocol with each searcher agent. The builder agent sends a Request message to the searcher agent asking as service to execute the action SearchWord, in other words, to perform the search in PubMed and subsequently to perform the analysis of the articles. When the searcher agent receives the Request message, it checks whether it understands the content and whether it can perform the required service (if the action that the message contains is SearchWord). If everything goes well, it sends an Agree message to the builder agent. Finally, when the searcher agent has made all its work, it sends an Inform message to the builder agent with the result of the action, which means, to build the ontology with the required information.

6 Conclusions

After making several tests and checking the obtained results, we can state that the objective of the partial project has been reached. It is possible to organize some reliable medical information existing in Internet with the help of the PubMed database. All tools used for information extraction and building ontologies have proven to be very useful, especially in the stage of experimenting and prototyping.

There are a number of issues that require further investigation. For example, the agents should be able to adapt and learn from the social interactions they experience. Agents should learn which acquaintances give reliable results in which circumstances. Based on this knowledge they should be able to adapt their selection appropriately. In case of Internet search, for example, they can learn the user profile and try to update previously retrieved information automatically without direct intervention of the user.

Acknowledgement

The work was supported by the research programme No. MSM 6840770012 "Transdisciplinary Research in the Field of Biomedical Engineering II".

References

1. Baujard, O., Baujard, V., Aurel, S., Boyer, C., Appel, R.D.: MARVIN, a multiagent softbot to retrieve multilingual medical information on the Web. Medical Information 23, 187–191 (1998)
2. Larsson, J.E., Hayes-Roth, B.: Guardian: An Intelligent Autonomous Agent for Medical Monitoring and Diagnosis. IEEE Intelligent Systems and their Applications 13(1), 58–64 (1998)
3. Decker, K., Li, J.: Coordinated hospital patient scheduling. In: Proceedings of the Fourth International Conference on Multi-agent systems, ICMAS-98, Paris (1998)
4. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley ACM Press, New York (1997)
5. Brewington, B., et al.: Mobile agents in distributed information retrieval. In: Klusch, M. (ed.) Intelligent Information Agents, pp. 355–395. Springer, Heidelberg (1999)
6. Gray, R.S., Kotz, D., Cybenko, G., Rus, D.: 'D'Agents: Security in a multiple-language, mobile-agent system. In: Vigna, G. (ed.) Mobile Agents and Security. LNCS, vol. 1419, pp. 154–187. Springer, Heidelberg (1998)
7. Menczer, F.: Complementing Search Engines with Online Web Mining Agents. Decision Support Systems 35, 195–212 (2003)
8. Nguyen, N.T., Ganzha, M., Paprzycki, M.: A Consensus-based Multi-agent Approach for Information Retrieval in Internet. In: Alexandrov, V.N., van Albada, G.D., Sloot, P.M.A., Dongarra, J.J. (eds.) ICCS 2006. LNCS, vol. 3993, pp. 208–215. Springer, Heidelberg (2006)
9. Dermott, P.M., O'Riordan, C.: A System for Multi-agent Information Retrieval. In: O'Neill, M., Sutcliffe, R.F.E., Ryan, C., Eaton, M., Griffith, N.J.L. (eds.) AICS 2002. LNCS (LNAI), vol. 2464, pp. 70–77. Springer, Heidelberg (2002)
10. Abasolo, J.M., Gomez, M.: MELISA. An Ontology-based Agent for Information Retrieval in Medicine. In: Borbinha, J.L., Baker, T. (eds.) ECDL 2000. LNCS, vol. 1923, Springer, Heidelberg (2000)
11. Rhodes, B.J., Maes, P.: Just-in-time Information Retrieval Agents. IBM Systems Journal 39(3/4), 685–704 (2000)
12. Lhotska, L.: Design of a Decision Support System in Cardiology. The 3rd European Medical and Biological Engineering Conference - EMBEC´05. Prague: SBMILI ČLS JEP, 2005, vol. 11, ISSN 1727-1983
13. FIPA: Agent Management. Available, http://www.fipa.org
14. Bellifemine, F., Poggi, A., Rimassa, G.: Developing multi-agent systems with JADE. In: Castelfranchi, C., Lespérance, Y. (eds.) ATAL 2000. LNCS (LNAI), vol. 1986, Springer, Heidelberg (2001)
15. PubMed. Available, http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed
16. Entrez. Available, http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html
17. HTMLParser. Available, http://sourceforge.net/projects/htmlparser/
18. Lucene. Available, http://www.apache.org/dyn/closer.cgi/lucene/java/
19. Snowball. Available, http://snowball.tartarus.org/
20. WordNet. Available, http://wordnet.princeton.edu/obtain
21. OWL. Available, http://www.w3.org/2004/OWL/
22. Protégé. Available, http://protege.stanford.edu/download/download.html

Knowledge Integration from Multidimensional Data Sources

Wilfried Grossmann[1] and Markus Moschner[2]

[1] WG Data Analysis and Computing, Dept. Computer Science,
University of Vienna, A–1010 Vienna, Austria
[2] WG Applications of Formal Logics, Dept. Computer Languages,
TU Vienna, A–1040 Vienna, Austria

Abstract. Information integration plays a substantial role for data warehouses with their needs for highly dynamical adaptivity. For data and knowledge integration an OLAP–based method is discussed with special emphasis on summary level integration. The basic OLAP–model is enlarged by a so called composite structure for documentation of the concepts behind the data according to the needs of statistical information processing. The model supports also bookkeeping of the evolution of the data model.

1 Introduction

In decision support one has to integrate many times information from different and usually multidimensional data sources. Such knowledge integration is quite common in the context of data warehouses, provided that all the necessary data sources are already represented in the warehouse data model. However, due to changing reality the definition of the data warehouse is not fixed for all times but needs often adaptation according to actual needs. Two well known examples are adaptation of data scope, if one wants to integrate new data segments into an already existing data warehouse, and adaptation of a warehouse scheme in order to combine loosely coupled datamarts.

Such types of adaptations, which occur also in connection with the setup of a data warehouse, have two aspects. The first one is the classical data modelling aspect, which means definition of a unified data model from two or more different models. The second one is the semantic aspect, which has to guarantee semantic consistency of the combined new data source. Typical examples of such semantic problems are questions whether the data refer to the same reality or whether the concepts and measurement methods producing the data coincide.

In case of adaptation of scope definition of the new data model is many times rather straight forward and requires only “simple” adaptation, or better alignment of the two data models. This problem has been considered in many papers and a number of tools are available for supporting such activities (for example Melnik [6]). In case of adaptation of the warehouse scheme we have to define a new data model compatible with the existing data models. And we have to check semantic consistency of the data combined according to this model.

The paper proposes a methodology for adaptation of data models in summary level integration of different sources, which considers the problem at the data level as well

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 345–351, 2007.
© Springer-Verlag Berlin Heidelberg 2007

as the data semantics level. Section 2 outlines the method for data integration and in section 3 we discuss issues of corresponding knowledge integration from a statistical perspective. In section 4 we conclude with some remarks about future work.

2 Data Integration

2.1 The Composite OLAP–Object Model

In practice one often has to combine various pieces of knowledge represented as data with different structures. Hence data integration needs a rather flexible data model which allows processing as well as retrieval of different types of data, which range from light aggregated transactional data up to highly aggregated tables represented in data marts. A convenient model for such data is the so called composite OLAP-Object data model defined recently by Pourabbas and Shoshani. We sketch only the basic features of this model, for details we refer to [8], [9]. Main constructs are the following ones:

- Classes, which may be either primitive classes that represent a finite enumerated set of values used as identifiers of the individual objects, or composite classes defined from two or more already existing classes.
- Attributes associated with the classes.
- Associations allowing pairing of classes. Of particular interest are one–to–many associations defining a hierarchy between objects and allowing summarization of objects with respect to summary attributes like count, sum, min, max, or average.

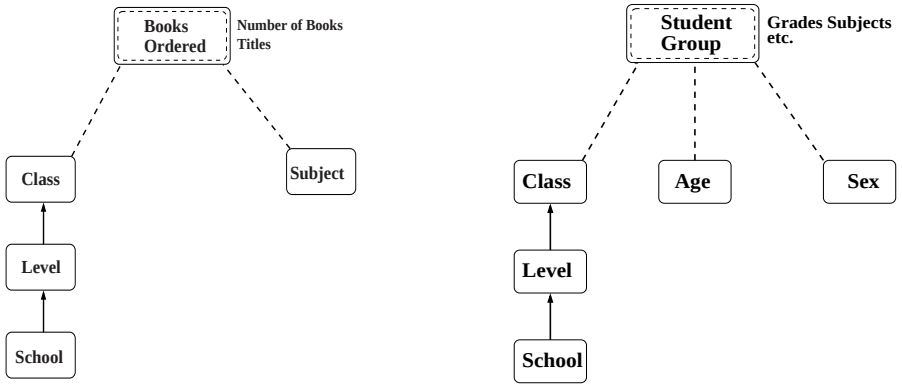
Let us consider the following example for illustration of the data model as well as the integration methodology. Suppose that a school administration maintains three databases:

- A detailed student database informing about grades of the students in different subjects;
- A teacher database containing information about teaching the different subjects in miscellaneous classes;
- A book database informing about the books ordered for the different subjects in miscellaneous classes.

Figure 1 sketches the structure of the student database and the book database and figure 2(a) the structure of the teacher database. Primitive classes are represented by simple border strips and composite classes are represented with double strips. Dashed lines indicate the connection of component classes with composite classes, and arrows show the associations defining hierarchies.

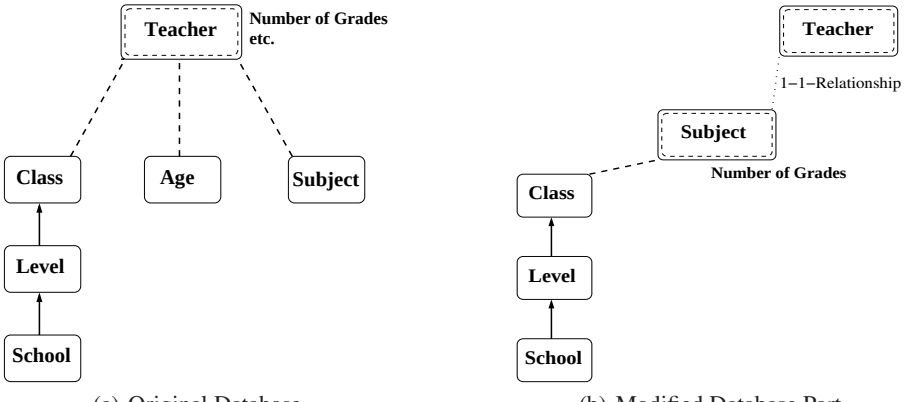
2.2 Integration Methodology

Integration of information from different databases needs a matching unit to which the different pieces of information refer. In terms of the Composite OLAP-Object model a natural candidate is a class occurring in both data models and summarization with



(a) Book Database (b) Student Database

Fig. 1. Basic Structure of Book and Student Database



(a) Original Database (b) Modified Database Part

Fig. 2. Basic Structure of Teacher Database

respect to this class is possible in both models. In context of the composite OLAP-Object model such a class is called an anchor. From semantic point of view an anchor represents a universe for which we want to obtain information. From database point of view the anchor is the focus of a query. It is defined either by a composite class or by one or more primitive classes. Integration of the data models along an anchor depends not only on the existence of such a class but also on structural properties of the data model with respect to the anchor. A straight forward solution is possible if summarization of the two data models with respect to the anchor is possible. A prerequisite for such summarization is that there are only one–to–many relations between the anchor and the other objects, and that the hierarchies for the anchor class coincide. In such case that anchor is defined by the finest primitive class within the hierarchy and the most informative dimensional model is defined by splitting all summarizable attributes

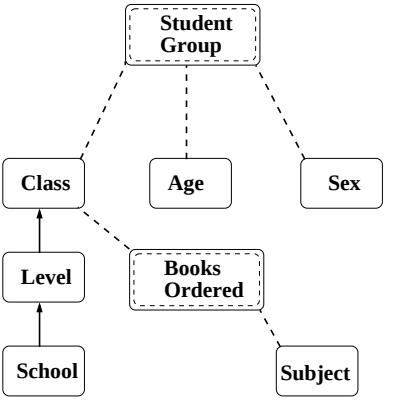


Fig. 3. Composite Structure of Book and Student Databases

according to the dimensions not used. For example, in case of the students database and the books database shown in Fig. 1 *School*, *Level* and *Class* are the common primitive classes with a hierarchy between. *Subject* is an extra primitive class for the ordered books, whereas *Age* and *Sex* are the extra ones for the second model. *Books Ordered* and *Student Group* are the composite classes. According to the rule stated above the anchor is defined by the object *Class* and we obtain a data model as shown in figure 3. This is the most informative model with respect to integration keeping for each class of students summary information about grades in the subjects, broken down by age groups and sex. With respect to the books ordered we keep summary information broken down by subjects.

If summarization with respect to an envisaged anchor is not possible then alignment of the structures is necesary. Hereby the existence of a viable alignment is assumed. Basic types of alignment operations may be listed according to the type of conflict resolutions:

- conflicts due to mismatch in one–to–many relationships;
- conflicts according to non corresponding value domains;
- conflicts due to mismatches in hierarchies caused by coarse refinements.

Usually such alignment cannot be obtained in automatic way but needs careful analysis of the underlying models. For example, in case of integration of the student database (figure 1(b)) and the teacher database (figure 2(a)). One cannot use the anchor *Class* directly because there is a many–to–many relationship between *Class* and *Teacher*. Hence summarization is not possible. Moreover *Age* in the teacher database has a different semantic as *Age* in the student database. A possible solution is transformation of the data model of the teacher database into a new data model by dropping all the information about the teacher and using only the objects *Subject*, *Class*, *Level*, and *School* together with the number of grades as shown in figure 2(b). This model can now be combined with the student model using the anchor *Class*. Using such a modification the integration of information from the student database and the teacher database can be done. The new model is similar to that one in figure 3.

3 Knowledge Integration

3.1 Basics

Up to now we have considered only the formal aspects of data integration by adaptation of the data model. However, such integration is only meaningful if we can ensure some kind of semantic coherence. Main issues of this semantic coherence in summarization are questions about comparability of the underlying universes to which the data refer. The universes must be the same for both data models and moreover coverage of the data with respect to this universe must also be the same. Such questions are not answered by the data model but need additional considerations. Statisticians have developed a peculiar view onto such problems and special so–called metadata models ([2, 7]) are available for alignment problems.

Packing such knowledge into a data model can start from general principles for structuring statistical knowledge using the main categorical concepts of statistics. Such basic statistical concepts are constituted by *dataset, (statistical) population and (statistical) variable*, which also constitute basic information concepts. The population and the units represent the universe from which the data stem from. This universe is a set of units, which are the carriers of the information contained in the dataset. Populations contain the real world objects. And the anchor classes refer just to these objects. Hence populations carry the corresponding class concepts. While units refer to primitive classes, populations refer also to composite classes in the sense of the OLAP–model.

The dataset itself is built of a number of variables, which represent the operational device for measuring the properties of interest for the units. In that sense the dataset represents instances according to the OLAP–model.

The information unit additional attributes are used for capturing results occurring in connection with data obtainment and computations. In context of data integration this information refers to the datasets used in the integration process.

The variables represent subject matter concepts such that these refer to OLAP–classes and contain attribute information, too. Variable domains represent possible results of the measurement process dependent on the setup of the measurement device.

Figure 4 gives some intuition on the connections between the fundamental concepts.

3.2 Knowledge Representation

In order to represent that part of data semantics which is beyond the data model itself we use an approach for describing metadata for statistically oriented summary data described in Denk et al. [3]. A so called composite structure, which pools together all the necessary information about the entities within categorical directories, is an appropriate and simple way to do this. Basically a composite consists of a number of data buckets described by a corresponding bucket scheme. This approach treats the main categories of metadata necessary for describing semantics shown in figure 4. Within such a model one can represent an essential part of data semantics necessary for meaningful data combination of the types occurring in above examples. Moreover the approach allows also checking feasibility of the operations with respect to this semantics — decidability and feasibility will be available for many practical purposes. The case of data combination considered in the first example is treated in some detail in [5].

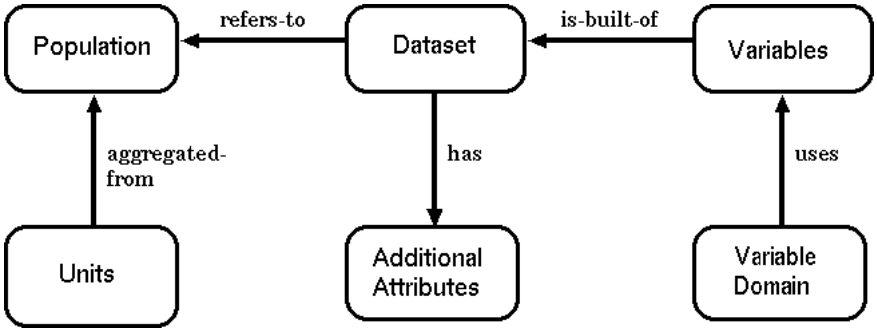


Fig. 4. Information units in connection with datasets

Because of their context knowledge friendly definition composites are well suited for bookkeeping of evolutionary changes. Hereto belong changes of anchors and corresponding populations as well as variables. Since anchors are connected with population concepts, semantically correctness conditions could be checked by use of the metadata information within the composites.

The application of OLAP–models with anchors opens the door to new methods in summary level integration. Slight changes in summary concepts may correspond with simple changes with respect to calculation methods. One example could be the consideration of simple exclusion conditions. Such information is available from accurate metadata descriptions in buckets. Common knowledge representation formats give no troubles for maintenance of such cases. Though more mathematical enhanced context needs new techniques of combination (well, representation of mathematical knowledge is already sufficiently achievable).

Another issue is the bookkeeping of evolution with respect to data models, variables and population. Basically such bookkeeping has to rely on the development of the anchors from already existing anchors or populations. Composites bring adequate structures for such bookkeeping of the anchors by definition of directories for each main category and storing in that way the evolution of the data model. In some sense this resembles the ideas of citation databases [1]. That counts not only for extra conditions mentioned above, yet also for tracking or comprehending historical or methodical changes. One should not forget that composites contain information about data structures, too.

4 Conclusion and Outlook

In this paper we considered a basic method for knowledge integration from different data sources, based on the computation of summary data for appropriately defined new units of information. The composite OLAP–model offers a well suited data model for such purposes; in particular the concept of an anchor is very useful for formal specification of data integration and supports gluing of information from different databases. Checking of semantic correctness requires a rather detailed description of knowledge

about the data, in contrary to traditional record matching. Representation of such knowledge requires an augmentation of the OLAP–model by statistically motivated description categories. Using a composite structure allows simultaneous processing of data and knowledge integration as well as bookkeeping of the evolution of anchors. The proposed approach is at the moment mere a heuristic for processing. A possible direction for future research is a more formal specification of the procedures in the direction of data, schema and ontology integration similar to [4].

References

[1] Buneman, P.: How to Cite Curated Databases and How to make them Citable. In: Froeschl, K.A., Grossmann, W. (eds.) Proc. 18th Int. Conf. on Scientific and Statistical Database Management– SSDBM 2006, pp. 195–203. IEEE, Los Alamitos

[2] DDI Data Documentation Initiative— A Project of the Social Science Community– Codebook, http://www.icpsr.umich.edu/DDI/codebook/index.html

[3] Denk, M., Froeschl, K.A., Grossmann, W.: Statistical Composites: A Transformation Bound Representation of Statistical Datasets. In: Kennedy, J. (ed.) Proc. 14th Int. Conf. on Scientific and Statistical Database Management, pp. 217–226. IEEE Los Alamitos, Los Alamitos, California/USA (2002)

[4] Goguen, J.A.: Data, Schema and Ontology Integration (Extended Abstract). In: Carnielli, A.W., Dionísio, F.M., Mateus, P.C. (eds.) Proc. of the Workshop on Combination of Logics CombLog'04, CLE e–prints, vol. 4(5) (2004), http://www.cle.unicamp.br/e-prints/vol_4,n_5,2004.html

[5] Grossmann, W., Moschner, M.: Towards an Ontology for Data in Business Decisions. In: Karagiannis, D., Reimer, U. (eds.) PAKM 2004. LNCS (LNAI), vol. 3336, pp. 397–407. Springer, Heidelberg (2004)

[6] Melnik, S.: Generic Model Management. LNCS, vol. 2967. Springer, Heidelberg (2004)

[7] The MetaNet Project, http://www.epros.ed.ac.uk/metanet

[8] Pourabbas, E., Shoshani, A.: The Composite OLAP–Object Data Model. Technical report, Lawrence Berkeley National Laboratory (2005), http://www-library.lbl.gov/docs/LBNL/592/29/PDF/LBNL-59229.pdf

[9] Pourabbas, E., Shoshani, A.: The Composite OLAP–Object Data Model: Removing an Unnecessary Barrier. In: Froeschl, K.A., Grossmann, W. (eds.) Proc. 18th Int. Conf. on Scientific and Statistical Database Management– SSDBM 2006, pp. 291–300. IEEE, Los Alamitos (2006)

Evolution of the Design and Implementation of Tutor: A Web-Based Educational System for University Courses

Roberto F. Arroyo, Miguel J. Hornos, and Rosana Montes

Dept. de Lenguajes y Sistemas Informáticos, E.T.S.I. Informática y de Telecomunicación, Universidad de Granada, C/ Periodista Saucedo Aranda, s/n, 18071 Granada, Spain
{robfram,mhornos,rosana}@ugr.es

Abstract. We have designed a web platform to support the teaching and learning of certain subjects taught in the University of Granada. This platform facilitates students and teachers diverse educational tasks and provides an effective management of academic data. Originally, it was intended for a reduced array of subjects, but as a result of its increasing success and use in a greater number of subjects, we have been encouraged to extend it. However, the approach used in the early design was a classical life cycle and did not take into account either future needs or its hypermedia system nature, which is even more important and requires a special processing, which is different from the one required for conventional applications. From our experience, we have learned and have taken into account the detected lacks to make the platform progress, and we have provided a new design thus changing completely its philosophy, structure and interface. We will hereby describe the main characteristics of both designs (the first and the new one) in order to highlight the evolution of the system.

1 Introduction

New technologies have extended to the global environments of our society, facilitating many activities and admitting tasks that were unlikely years ago. In education, the teaching staffs have recognized the need for effective tools to manage information resources for their students' progress and for their subjects. The European Space of Higher Education [1,2] promotes the employment of new technologies in order to increase the cooperation among universities from its member countries and improve the educational process quality in them. In recent years, and in the context of this new educational framework, the development of electronic communication channels has allowed higher levels of cooperation within the educational community. Nowadays, computers are everywhere and university students are able to communicate no matter where they are. ICT (Information and Communication Technologies) as the World Wide Web [3] have become one of the main mechanisms for remote interaction, and have reshaped both society and universities all over the world. Universities must capitalize on the web for teaching and learning, and the increasing use of web-based systems is a progressive manner to be on the way.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 352–359, 2007.
© Springer-Verlag Berlin Heidelberg 2007

According to these needs, the University of Granada has created several plans of educational quality [4,5]. Within them, we are developing a teaching innovation project with the purpose of building a web-based platform, called Tutor (http://tutor.ugr.es), which provides a toolkit to support the educational activities of both teaching staff and students.

This web-based educational system, which is particularly intended for university courses, must provide an adequate pedagogical approach in order to increase its usefulness and achieve the following main proposed objectives:

- To provide additional information focused on academic contents, such as subject syllabus, didactic material, glossaries, learning activities, etc.
- To establish new communication and interaction channels among the different academic members.
- A better management of all registered users provide a secure authentication mechanism to preserve the users' identity and privacy of their data.

The system we developed in the beginning was adequate for a limited number of users and subjects, but due to its design, its adaptation to such a big growth (in users, contents and specially in functionality) requires a disproportionate effort. This mainly occurs because the hypermedia nature was not taken into account; thereby, the information storage was not separated from navigation aspects; a formal user modelling was not performed and mechanisms to carry out a satisfactory adaptation were not defined. For all these reasons, we have decided to develop a new version of Tutor, which, as well as improving the user interface and navigation system, admits adaptiveness and adaptability.

The rest of the paper is organized as follows: Section 2 will present the context and design structure regarding the first implementation of the platform. Section 3 will show the architecture and design principles followed in the development of a new version of the system. Finally, in the last section the conclusions will be outlined.

2 Initial Design

The platform initially designed follows a role-driven approach, using a role-based model. Firstly, the system roles are distinguished and the different parts that made up the whole system are developed. Next, we establish relations between the roles and parts using their operations, which will be implemented on a later developing stage. For operation modelling we use a user-driven design, which provides a higher abstraction level for system task description and separates the concepts of users and roles.

Using a role-driven approach allows us to get different functionalities according to the role played, which is reflected on the user interface. The system distinguishes between four different user profiles (see Figure 1): administrator, teacher, student and unauthenticated user. When users log in the system, it automatically recognizes the roles they play. Once access has been granted, the system displays a menu with the sections and actions allowed in conformity with the role the user is playing.

The system is based on a client/server [6] approach with dynamic content generation. It uses Apache [7] as a web server of dynamic pages, which is

accompanied by MySQL [8], a relational data base used for data storage. A collection of PHP [9] scripts implements operations and generates the HTML interface. This part is server side, both for storing and executing. We use a Sun server to support many simultaneous connections. As a client, the user can use any HTML[10]/CSS[11] compatible web browser to access the platform from any place with an Internet connection. In brief, data manipulation is not separated from the own functionality of the application, so that queries and modifications to the database contents are carried out together with the code processing, thus adding complexity to the code updating and data maintenance.

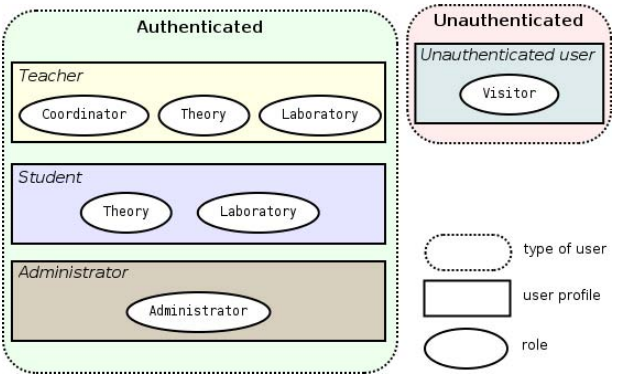


Fig. 1. Structure of user profiles and roles

The main problems presented in this implementation are the lack of some important features. Among the latter we can highlight the following ones:

- The whole system is short of friendly navigability. A series of menu items and information nodes are not easy to find, and many clicks are needed to access some of them.
- It has not a clear look-and-feel interface. The appearance of the system is neither very elegant nor intuitive.
- The implementation is not modularised, since the same or similar functionality is reimplemented in several fragments of code. This increases the number of programming errors and makes its maintenance difficult. At the outset, every role had its own functions, but several roles had the same functionality. Since every role has its own implementation, then, for the same functionality, there are different implementations in the system. Moreover, the latest implementations consider all the roles sharing the same implementation: internally, the function determines which portion of specific code must be executed according to the role played by the user. All this leads to a non-homogeneous implementation.
- The system does not support adaptability or customisation. Different users cannot have different interfaces.
- It is quite difficult to extend the system with new functionalities without breaking the system. It has not been designed to be extended easily.

- The maintenance of the system is hard. Due to some of the above mentioned items, fixing problems in the whole platform is a hard work.
- It does not follow the XHTML standard [12], using only the HTML document definition.

3 New Design of the Platform

The motivations for a new design from scratch are to solve the lacks of the earlier model while maintaining the same goals of the previous version of the platform. Therefore, we are looking for a 24/7 web-based system to support educational tasks and academic data management that provides additional student-student, teacher-student and teacher-teacher communication channels. Basically, the differences were conceptual, so that we do not want to ignore its hypermedia nature, developing it under the object oriented paradigm, separating contents from presentation and using an implicit role assignment method. With all this, we pretend to avoid and solve the most important problems of the previous version, namely:

- A greater number of users requires different and new functionalities.
- Some users get lost trying to use some of its current applications.
- The maintenance is expensive due to a redundant code.
- The navigability should be more intuitive and direct (with less mouse clicks).
- The system should be scalable and future-prone.
- The new XHTML/CSS technology should be used.
- It would be interesting to provide a multi-language platform, since the mother tongue of many users (e.g. foreign students) is not Spanish.

Tackling these main goals and features, we have designed the new model for the platform implementation.

3.1 Design Philosophy

The previous design was oriented to the role (or user profile), directing both the way in which the data were stored and the form of presentation and navigation to the role played by the user navigating in the system, independently from the part of the system being visited. The new design completely changes this orientation, proposing an object oriented design (data and operations), where what is visited, and not only who visits it, is also important. The first-level objects we have considered include the system users, subjects, degree courses, groups, faculties and departments, as well as notices and internal messages. All these objects constitute the system navigability core, and are the main information structure of the system; hence, their design deserves a major consideration. The new design is divided into two layers (see Figure 2): data storage and adaptation to each user. The connection between them is the user model, which stores, maintains and updates the user's main features.

As we have already mentioned, one of the main objectives of the new design is to separate contents from presentation. This admits multiple presentations (taking into account look-and-feel, language, restrictions, etc.) associated with a given piece of contents. In this design, the data is stored in a relational database, and the system queries the database to display the contents. Therefore, the data storage layer converts

relational data into the object oriented structure to be shown. Later, the selected contents will be published according to navigability restrictions, user capabilities and preferences, and style templates.

Another important objective is modularity. Common functionalities must be implemented only once and be shared among modules. Additionally, all the database queries are encapsulated in a separate module, outside any other module. Finally, to make up the final appearance of the information, we use XHTML templates with pattern replacing.

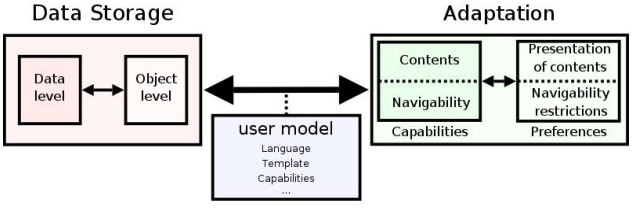


Fig. 2. New design of the system: two independent layers linked by the user model

3.2 Structure

The data storage layer contains the information. This layer is composed of two levels: the data level and the object level. The first one deals with the relational database, whilst the second is responsible for converting object-oriented queries into relational ones and relational data into object oriented structures.

The data presentation related to the personal and hypermedia criteria is carried out in the adaptation layer, which is made up of two different levels. The level of capabilities is responsible for determining which contents are displayed in accordance with the navigability patterns and with the relation with the user himself, e.g. a teacher has access to their students' e-cards, but not to those belonging to students that are not enrolled in any of his/her subjects. The level of preferences modifies the presentation and filters the contents in accordance with the user preferences and navigability restrictions.

Figure 3 shows the internal structure of both layers, and an example of the conversion of relational data into an object oriented structure carried out by the object level.

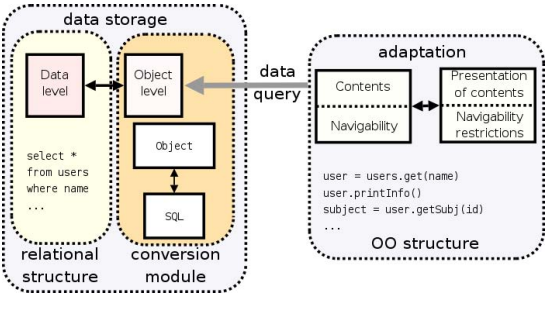


Fig. 3. Structure of the new model

3.3 Template System

The different presentations of the contents are prepared using a template system. We have defined the following four different template levels:

- *System template*, which is the first template the system uses. It shapes the main appearance and structure of the whole system, and determines which parts must be filled with next templates.
- *User model template*, which admits changing parts of the system template according to the data stores in the user model. For example, accessibility issues can be attended in this template.
- *Capability template*, which adapts the general appearance according to the user capabilities. This allows customizing the user interface and navigability of teachers, students, administrators, or the appearance of a determined object in relation to a specific user.

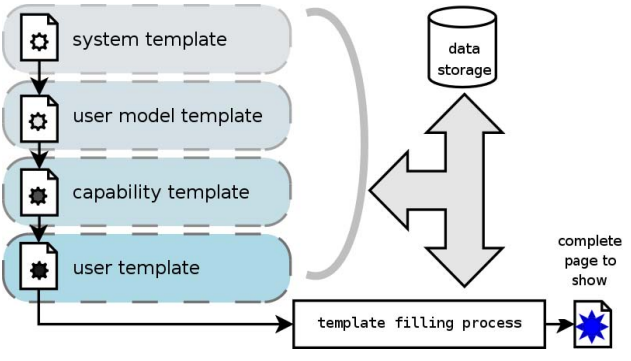


Fig. 4. Template processing: The final web page is generated by making up four different templates and accessing the database

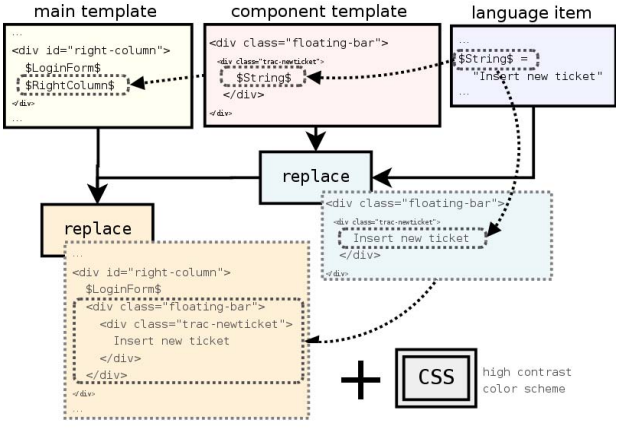


Fig. 5. Replacement of key values to convert templates into the final document

– *User template*, which lets the users customize their user interface. In such a way, every system user is able to choose certain aspects in order to customize the appearance of the contents presented by the system.

Figure 4 shows our sequential processing of the templates, from the system template to the final web page to be shown.

The final contents of the web page to be displayed are obtained by successively replacing key values (included in the different templates) with the real data, which could be from a simple string to another group of key values. The final document is obtained after all the key values are replaced. Figure 5 shows an example of this replacement.

4 Conclusions

We have presented how the design and implementation of a web-based educational platform has evolved from its initial version to the new one, in such a way that the latter constitutes an example of applying quality design principles to a hypermedia system. The evaluation of the first version of the platform showed that it lacked the essential basis to be easily flexible and adaptable to users, being necessary to spend a considerable effort to add new functionalities, as well as a more intuitive navigation system. We have also learned that the followed role-orientation, without taking into account the visited information, did not contribute to improve the system navigation.

The main problems detected in the previous version have been solved with the new design, while its positive aspects have been kept. The perspective change from role-orientation to object-orientation allows a more intuitive and controlled navigation, with a better structured presentation of the information. This admits an easier expansion and upgrading of the platform to new requirements and needs, in such a way that it is at the same time more scalable, and does not affect negatively the provided functionality. The active role of the system to make decisions has also been increased, by replacing the manual selection of the user profile with an automatic query to determine the user's capabilities while the user visits any page of the system.

The new design establishes two independent but interrelated layers: the data storage layer and the adaptation layer. The link between both layers is the user model. The data storage layer is structured in two levels: the data level and the object level. The former maintains a relational representation of the information and communicates with the DBMS, while the latter builds extended objects in accordance with the existing relations among data and communicates with the adaptation layer. The adaptation layer performs an adaptive process in two levels, deciding respectively what to show (i.e. the contents and their navigability) and how to show it (i.e. the presentation of these contents taking into account the navigability restrictions, which can hide part of the selected contents to this user). The first level depends on the user's capabilities, while the second one depends on the preferences established by each user.

The implementation of the new version of the platform is in an advanced state, but not finished, so we are currently working in order to finish its development, with the aim of replacing the previous version. We also plan to increase the functionalities of

the new system, and, at the users' request, extending the available tools with new options and operations and adding new tools to the ones already included in the platform.

Acknowledgments

This work is financed by an Educational Innovation Project from the University of Granada (code 05-03-46).

References

1. Ministers responsible for Higher Education: Realising the European Higher Education Area. Berlin, Germany: Official Communication (2003)
2. Treasury of Education, Culture and Sport: The integration of the Spanish university system in the European Space of Higher Education. White paper (in Spanish) (2003)
3. Berners-Lee, T.: Weaving the Web: The original design and ultimate destiny of the World Wide Web by its inventor. Harper Collins, New York (1999)
4. Vice-chancellor of Planning, Quality and Educational Evaluation, Plan of Educational Quality, /2004. University of Granada, Granada 2001 (in Spanish) (2001)
5. Vice-chancellor of Planning, Quality and Educational Evaluation: Plan of Educational Quality, /2008. University of Granada, Granada 2005 (in Spanish) (2005)
6. Edelstein, H.: Unraveling Client/Server Architecture. DBMS 7(5), 34–41 (1994)
7. The Apache Software Foundation, online at, http://www.apache.org/
8. MySQL AB, online at, http://www.mysql.com/
9. Welling, L., Thomson, L.: PHP and MySQL Web Development. Book & CD edn, Sams, Crawfordsville (2001)
10. W3C: HTML, online at, http://www.w3.org/html/
11. W3C: CSS, online at, http://www.w3.org/Style/CSS/
12. W3C: XHTML2 Working Group Home Page, online at, http://www.w3c.org/MarkUp/

Using Videogames in Special Education

José Luis González, Marcelino J. Cabrera, and Francisco L. Gutiérrez

Software Engineering Department. University of Granada,
C/ Daniel Saucedo Aranda, s/n, 18007, Granada, Spain
{joseluisgs,mcabrera,fgutierr}@ugr.es
http://lsi.ugr.es/~gedes

Abstract. Consideration has arisen on the use of video games as learning tools since they are deemed as active elements in classrooms. New game technologies have helped creating alternative strategies to increase cognitive skills in the field of Special Education. This paper describes our experience in video games design and in new forms of human-computer interaction addressed to develop didactic games for children with communication problems such as autism, dysphasia, ictus or some types of cerebral paralysis.

Keywords: video games, autism, interaction systems, learning tools, multiple intelligences, stimulus equivalences, communication systems.

1 Introduction

Nowadays the attention to people with special needs, in particular those with communication disabilities, is an area in continuous expansion in our society. Hence, it is very important to develop technological solutions for the rehabilitation and integration of people with communication difficulties, independently from their problems (paraplegia, cerebral palsy, autism, etc.).

The need of generating new communication strategies and learning methods in special education requires the development of original adaptive and personalized software and hardware tools. Video games are an attractive and direct platform to approach children. They provide interesting human-computer interaction methods to enrich the learning process in special education. Moreover, games help to improve social relationships, to raise the communication level and to ease the assimilation of new concepts that improve the learning process.

In this paper we will review the meaning of the concept *learning*, and how it can evolve to include games as tools that support pedagogic development. We will offer a vision of games as didactic tools and of how they should be used as instruments in the field of Special Education.

We will introduce Sc@ut DS, our new tool for learning communication concepts addressed to autistic children. It uses game concepts and video console hardware platforms.

Finally, we will comment on some rules that could be followed to develop didactic units, on games using the *Stimulus Equivalence Theory*, on new interaction modes that new game devices offer and on how to use them to help people with disabilities in the classroom.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 360–367, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 "Learning by Playing". Video Games in Special Education

Can people learn by playing? This is a question largely debated by specialists in Psychology and Pedagogy. Before answering this question, we should understand the meaning of Learning and Playing and whether these concepts can be combined to create new applied learning methods without forgetting that we are working in Special Education. Therefore, we ought to adapt the contents to the characteristics and restrictions of these groups.

2.1 Learning in Special Education

Learning can be defined as the form of acquiring knowledge by studying and gaining experience. The learning process involves being able to associate semantics or mental ideas with real world objects, establishing a relationship between the mental world and the real one.

Sometimes, traditional learning mechanisms make students loose interest towards the subjects. Both the studies and the lack of interest often make the learning process hard and difficult.

In Special Education the learning process is usually limited to cognitive weaknesses. It is essential to relieve students from the pressure exercised by traditional techniques and to look for new mechanisms that would help to increase both attention and motivation levels and to assimilate new concepts.

2.2 Video Games in the Classroom

We can define *playing* as the action or set of actions directed to having fun or merely to spending time. When playing, we follow a series of rules to solve a problem individually, or collaboratively, against other players or a machine. The main factor in a game is motivation. Thanks to motivation, players can be happy when they play.

A video game is a computer program specifically created to entertain. It is based on the interaction between a person and a machine. Here is where the video game starts running. These programs recreate virtual environments where players can control mainly characters or any other element to accomplish one or several goals by observing a set of rules.

Only in Spain, the video game industry is currently leading, in terms of turnovers, about 967 millions of euros in 2006 - Data published by Adese (Spanish Distributor's Association and Software's Entertainment Editors). Many specialist educators agree on the importance of video games in the present social context [1][2].

In the scholar context, video games have been traditionally considered as simple entertainment mechanisms to be used out of classrooms, thus missing their advantages as learning tools [3]:

– *Scholar success:* Pupils that have used video games, have increased their reading comprehension capability.

- *Cognitive abilities:* Pupils train these abilities using environments based on discovery and creativity.
- *Motivation:* Games are an encouragement mechanism for children, they make the learning process easier and increase attendance considerably.
- *Attention and Concentration:* Attention and concentration are incremented to solve concrete problems due to pupils' nature towards games.

2.3 Learning by Playing, Playing to Learn

After introducing some concepts about the learning process and their application in classrooms, we could wonder weather it is possible to apply them to Special Education too. When playing, we can create situations where we can examine our own limits in a self-regulated way. The game is the mediator of the learning process [4].

The use of video games in Special Education shows important problems:

1. Educational video games often are not designed for people with special needs. Few video games with educational contents may be used in this context.
2. The already existing video games for special education are mainly didactic units which have lost the essence and attributes of games.
3. The devices where these didactic games are implemented on are just simple PCs. They often do not raise children's interest.

Due to the above mentioned reasons, it is necessary to act carefully when transferring the experiences of games in the field of General Education to the area of Special Education . New design mechanisms must be proposed to develop this kind of games. This should enable people to overcome their cognitive deficiencies and to increase their social integration level by applying the suitable techniques during the design of these games [5].

In order to favour accessibility in Special Education we have used Gardner's "Multiple Intelligences" Theory [6]. This theory states that Intelligence is not a unitary and indivisible concept but a set of abilities or capabilities that can be trained (Fig. 1).

Video games can be very useful mechanisms to enhance different abilities. It is possible to use *other intelligences* to improve another *weaker intelligence* by creating worlds and situations that allow acting upon a subset of abilities to surpass the problems associated to others.

2.4 Video Games for Special Education

In this context, regarding the video games design process, not only it is extremely important to consider the characteristics of the games but also the qualities they should have as learning tools. These are some points to take into account:

1. To identify the player's profile, his/her limitations and cognitive capabilities in order to choose the best multi-modal interaction mechanisms.



Fig. 1. Multiple Intelligences and their relationships

2. To structure and adapt the didactic unit to the player's characteristics goals of the game. It is important to reach a balance between what we want to teach and what should be learned.
3. The educational contents should be introduced into the game structure in a hidden way. The game should have its objectives as a game, and special objectives as a learning tool. Hence, the child can learn indirectly while playing.
4. To evaluate the positive aspects that the game offers, without forgetting the negative aspects that can create vices or inappropriate conducts of the child.
5. The game must offer feedback for each action. This feedback is focused on the cognitive need which the child must train when playing. To develop this feedback we will make use of the other intelligences as supporting tools and also of the child's profile.
6. It is important not to create frustration in our players due to their disabilities. Errors should be corrected without causing sadness or discouragement.
7. To use a main character or hero, who acts as a guide in the learning process and game. Children should see themselves reflected in this hero.
8. Each game must have goals. The learning process should be in rise, based on multi-levels or missions where the level of difficulty increases gradually. At each level the cognitive weakness should decrease.
9. To give rewards for correct actions: animations, songs, videos, points, objects, and even, gifts in the *real world*.
10. The mechanism to carry out an action in the game should have the same structure as the mechanism that would be used to solve it in the *real world*.

3 Sc@ut DS: Playing for Communicative Learning

As an introduction example of video games as special education tools, we have developed Sc@ut DS [7]. This project is an evolution of another project called

Sc@ut and is an alternative and augmentative communicator (AAC), developed by the research group GEDES of the University of Granada using PDA devices as hardware platforms.

We have observed a series of deficiencies in PDAs during the use of this communicator:

- *Fragility:* The touch screen is very delicate, getting scratched or breaking easily.
- *Autonomy:* About 3 hours. The battery requires replacing several times a day, being this aspect a great inconvenience for children and tutors.
- *Multimedia:* The main memory is limited and consequently also the multimedia objects and user interfaces which make them not attractive to children.
- *Price:* High, around 300€, being a negative factor for parents and tutors.

The main goal of this project was to obtain a new attractive platform, and a less limited than Pocket PCs for children and to change the communicator's philosophy to develop a learning tool using game concepts.

To develop the new communicator we have selected Nintendo DS™instead of other alternatives such as PSP™(Playstation Portable) or UMPCs (Ultra Mobile PCs), because we think it has the following advantages:

- *Feedback:* It has two screens. One of them is a touch screen which offers more interaction possibilities.
- *Multimedia:* It's a game device with great multimedia options (sound, video and graphics) without apparent limitations of memory.
- *Autonomy:* Approximately 11 hours.
- *Connectivity:* Wireless (Wi-Fi), to communicate with other DSs and PCs
- *Price:* About 150€, much more affordable than a PDA device.
- *Durability:* It is a device designed to be used by children
- *Motivation:* Other commercial games can be used as a complement, gift o reward for children.

The main didactic goal is to increase interaction and social communication. Children learn how to make sentences to express their desires and feelings. The device makes the learning process easier by imitation.

We use both screens to show an adequate feedback, using videos and animations without losing the context of the action, and showing jointly the cause and the effect of the action (Fig. 2). Concepts are represented by pictograms whose designs are personalized according to the ideas of the personal world of each child. In order to improve the grammatical language in communication, we show the syntactic path to be followed. In this way, children can learn how to communicate using a simple structure of pictograms which represent the mental idea they want to express.

We have created a series of animations. Thus, children can correlate the grammatical structure with the mental idea in a entertaining way, and finally they can associate it with the correspondent action in the real world.

To enrich oral communication, the device reproduces an audio file that expresses the concept selected by the child. When he/she finishes a sentence Sc@ut



Fig. 2. Cause and Effect represented jointly on Sc@ut DS

DS shows the complete pictogram path and reproduces the full oral phrase. Our hero is "Leoncio, The Lion". He is the mediator in the learning process and the one who carries out the actions following the pictograms' path. He's very expressive and helps to obtain a positive action assimilation.

4 Didactic Videogames in Special Education

When didactic games for special education are developed, psychological techniques must be used to help to overcome the users cognitive limitations. In our context this limitations are autism and cerebral palsy. We apply Sidman's theories about "Equivalent Relationship" and more explicitly the "Stimulus Equivalence" mechanism [8]. This theory concludes that if relationships between stimuli are trained clearly, then new relations that have been hidden can be obtained implicitly. The relationships between stimuli are: Reflexivity, Symmetry, and Transitivity.

Our goal is that children learn how to associate letters, words or sentences to concepts. These concepts could be pictograms or videos that represent the action, with the correspondent oral pronunciation.

At very advanced stages, one of our didactic games for people with autism is a about Food concepts. A goal is to correlate food pictograms with the written word with direct stimulus training. On the other hand, indirectly, when a child listens to the oral pronunciation of the concept (pictogram), he could implicitly associate this sound with the written word by "Stimulus Equivalence" and by the *Transitivity* property (Fig. 3). In the game structure we use Discrimination with Minimal Error techniques. With correct stimuli we introduce erroneous stimuli where the child must select to obtain the correct result. The concepts location on the screen must be at random in each level. Thus we avoid children always selecting the same position (pictogram). If the child chooses the right option, we show a positive feedback (animation o video) focused on the concept of learning and therefore we improve the learning by imitation. But if a child fails, we remark that the right concept and the pictograms are randomly redrawn at different locations. It is important not to forget that it is a game and that a score is needed. When the score is high enough, the child can obtain new levels, exercises and rewards. This is a relevant aspect in order to improve the child's

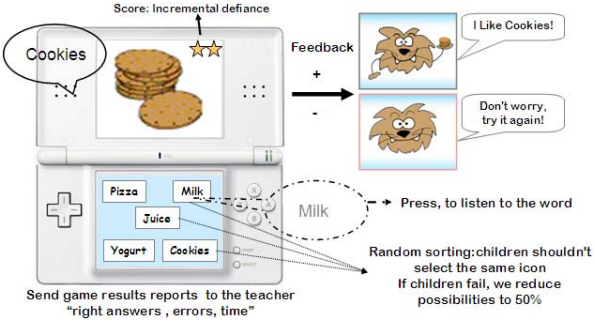


Fig. 3. Sc@ut DS game prototype with "Stimulus Equivalence" structure

motivation. Finally, we can send the game's result to the teacher as a complete report using the wireless connection. This report can be used afterwards to gather information of several aspects of the learning process of any child.

To overcome the small size of the touch screen, for some children we use a Wii remote control (Wiimote) as a new interaction mechanism. This control detects the user's movements and position in the 3D space similarly to a three-dimensional mouse. With this control, we adapt games to be used employing a television or PC monitor as screen (Fig. 4).



Fig. 4. Alternative interaction in Sc@ut DS using Wiimote

5 Conclusions and Future Works

In this paper we have showed our studies about using video games as a learning tool in special education. We propose a series of guidelines to design special education games and use video consoles as alternative devices since they are specifically designed for playing, as well as very entertaining for children and offer better human-computer interaction adapted to certain collectives with disabilities. Examples of these premises and methodologies in this work have been shown with Sc@ut DS, our communication learning tool alternative to Sc@ut.

Finally, we conclude that videogames in special education offer very interesting results:

- Better spatial, temporary and hand-view coordination ability.
- Better concentration, motivation, attention and reasoning capability.

- Better assurance in the learning process. Children can repeat a task until they dominate it.
- Better assimilation of strategies an consequences in determinate actions.
- Children are happy playing and learning. This improves the social relationships with other children and the environment.

A future goal is to generalize the development process we are following and propose a complete methodology to create and design adaptative and personalized games based on users profile in order to help in the social integration and the learning process of people with disabilities such as autism. We are currently developing new didactic games for other special education disciplines and the introduction of new alternative learning methods as symbol languages for deaf-mute or autistic children. We will begin an experiment in the next month on a group of children to check the efficiency of our methods.

Acknowledgments. This study and work is financed by the *Consejería de Educación (Education Department) de la Junta de Andalucia (of the Council of Andalusia)*, as a part of SC@UT Project, the Spanish International Commission for Science and Technology (CICYT) and AMENITIES project (TIN2004-08000-C03-02).

References

1. Provenzo, E.: Video kids. Harvard University Press, Cambridge (1991)
2. Greenfield, P.M.: Video Games as Cultural Artifacts. Interacting with Video. Ablex Publishing, Norwood, NJ (1996)
3. McFarlane, A., Sparrowhawk, A., Heald, Y.: Report on the educational use of games: An exploration by TEEM of the contribution which games can make to the education process (2002), http://www.teem.org.uk/publications/teem_gamesined_full.pdf
4. Rosas, R., Nussbaum, M., Grau, V., López, X., Salinas, M., Flores, P., Correa, M., Lagos, F.: Más allá del Mortal Kombat: Diseño y evaluación de videojuegos educativos para Lenguaje y Matemáticas para el Nivel Básico I. Psykhé, 9. pp: 125-141 (2000)
5. IGDA. Accesibility in Games: Motivations and Approaches. International Game Developers Asscoiation (2004), http://www.igda.org/accessibility/IGDA_Accessibility_WhitePaper.pdf
6. Gardner, H.: Frames of Mind: The Theory of Multiple Intelligences. Basic Books. ISBN 978-0465072510
7. González, J. L., Cabrera, M.: Sc@ut DS: Sistema de Ayuda a la Comunicación y al Aprendizaje. IV International Conference on Multimedia, Information and Communication Technologies in Education (m-ICTEE2006). ISBN: 84-690-2472-8. pp: 1116-1120 (2006)
8. Sidman, M.: Reading and auditory-visual equivalente. Jounal of Speech and Hearing Research 14, 5–13

Facing the Challenges of the European Higher Education Area: The dMath Project

A. Martín[1], G. Rodriguez[2], and A. de la Villa[3]

[1] Universidad de Salamanca. Dep. de Matemática Aplicada.
E.P.S. de Avila. Hornos Caleros 50, 05003
Avila (Spain)
delrey@usal.es
[2] Universidad de Salamanca. Dep. de Matemática Aplicada.
E.P.S. de Zamora. Avda. Requejo 33, 49022
Zamora (Spain)
gerardo@usal.es
[3] Universidad Pontificia Comillas. Dep. de Matemática Aplicada
y Computación. ETSI (ICAI). Alberto Aguilera, 23. 28015
Madrid (Spain)
avilla@upco.es

Abstract. Adaptation to the structure of the different University studies to the European Higher Education Area (EHEA) implies a deep reform affecting the structure of University studies. A brain storming about the mathematical courses will also be analysed.

After some considerations about e-learning and b-learning we will describe the main features of the project: *Building a European Database of Mathematical e-learning Modules*: dMath. We pay attention to the general characteristics of the project and the different modules comprising it.

1 Introduction

The tailoring of the structure of different university studies to the European Higher Education Area (EHEA), an objective forthcoming from the Bologna Declaration, is the major task outstanding in Europe's different university systems. The reform mainly affects the structure of university studies. We also need to reflect about the suitability of the syllabuses in mathematics subjects and the way in which mathematics is taught. It is now no longer possible to uphold the same approach to teaching mathematics as fifty years ago and classrooms should reflect the technological revolution that has occurred in recent decades.

In our opinion, this situation takes place especially in the teaching of mathematics in Engineering Schools, where they are characteristically applied mathematics. Therefore, they turn them into an indispensable tool to acquire the different technical competencies necessary for European Engineers. This requires the drawing up of a new high quality didactic material including theory, problems and applications the professor may use for individual teaching in the different areas of learning demanded by the new situation.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 368–375, 2007.
© Springer-Verlag Berlin Heidelberg 2007

We shall present a new tool, forthcoming from the European project **Building a European database of mathematical e-learning modules (dMath)** [3,4], within the Leonardo Da Vinci Programme (second phase: 2000-2006), which will undoubtedly be an extremely useful tool in favour of the review of content and methodology that European society requires.

The Bologna Declaration makes clear statements about quality assurance in EHEA, being the quality of teaching a part of it. Essential objectives of EU programmes into Leonardo and Minerva are in accordance with this and new teaching methods and tools will be important. The Commission has approved dMath within this context and dMath will undoubtly be important as it comes to increasing quality in engineering mathematics teaching.

Before dealing with the description of the new project, it is necessary to comment briefly on the substantial changes that the new teaching scene is going to provoke in the teaching of mathematics in the Engineering Schools. It is in the context described below that the working out and the use of the didactic material acquires an especial sense.

2 The European Higher Education Area

In order to help define what this new paradigm of European university teaching means, we must first outline certain features of the European area that are currently being designed [6,9]:

- A competency-based teaching offer.
- A diversified teaching offer (theoretical and practical teaching, supervised academic activities, independent individual work and so on)
- Concern for the students' overall work, as opposed to the current system in which the only measurement in certain European countries is the hours of class. This work should have a maximum annual volume estimated at 60 European credits (European Credit Transfer System, ECTS). An initial estimate suggests that 1 ECTS credit is equal to 30 hours of student work, including class attendance, laboratories, workshops, individual and group tutorials, individual or group work and assessment.
- Prevalence of student learning over the instruction (face-to-face, distance learning, blended learning) provided by the teacher.
- Duration of studies that is more in keeping with reality.

The demands on the teaching staff, stemming from the European-wide policy on harmonising higher education studies, are basically as follows:

- Teachers should teach things of value to the labour market (competencies) and adopt a different approach to teaching (methodological innovation).
- The basic criteria for the planning and development of teaching should be students and their needs (students as a referent).

All of this involves a greater dedication from the part of teachers to a student's learning process (teaching hours, material's preparation, suggested lines of work, guidance and supervision, etc.).

3 A New Teaching Scenario for Mathematical Education

In view of the characteristics of the EHEA and the aforementioned demands on teachers, certain changes, sometimes of a drastic nature, are required in our traditional approach to teaching. The changes required do not only affect syllabuses, stemming from weaker initial grounding, but, in general, affect the knowledge of mathematics acquired in the stages leading up to a university education. This is highlighted by different European-wide reports, such as the PISA report, and there is also a need to embrace a methodological change that which will allow addressing the students' new learning needs. The use of the entire potential provided by new technologies applied to teaching may lead to reconsideration of the organisation of different teaching groups, adapting their size to the possibilities of the different computer rooms [11,12].

Finally, we should change the assessment process, seeking mechanisms that assess the whole learning process. New technologies are once again the key instrument, allowing the introduction of well-designed self-assessment processes, guided practical sessions, etc.

This whole situation requires teachers to take special care when planning the following aspects:

- The competencies that students will acquire individually within the context of the total offer of competencies that are taught in the subject in question.
- The academic scenarios in which the teacher wants the learning to take place, referring to specialized classrooms (computer rooms, laboratories, audiovisual rooms, etc.), to libraries and other documentation centres, provided that the students can proceed independently within them. In addition, the proper use of e-learning can contribute to the learning process.
- The working process to be followed by the student for the acquisition of these competencies, detailing the stages that constitute the overall work sequence and the tasks to be performed by the student and by the teacher during each one of the stages and estimating the time each one of them will take (a suggested number of hours and calendar).
- The system of tutoring supervision that the teacher has adopted for controlling the acquisition of competencies on the part of the students. This implies that we should specify at least three aspects:
  - The format of the supervision (face to face, e-learning and b-learning).
  - The moments when the student should approach the teacher to report on the progress made (number of tutorials to be attended).
  - The access to the website or e-mail contact and the dates upon which they are to take place.

The features and requirements of the supervision (what the student has to submit and how, written assignments, etc.)

The system for assessing the competencies acquired by the student on an independent basis. Assessment criteria should be put in place accordingly (results expected in each one of the stages and overall results of the process) and the

evidence regarding each criterion (data collected, reports generated, etc.) liable to assessment, the assessment tools and the assessment dates.

The subject's syllabus should therefore specify all these aspects of the tutored learning offer provided by the teacher.

The teacher will be required to provide students with material, give classes, set supervised work or problems, arrange different kinds of tests, use the latest technologies to a greater or lesser extent, etc., in order to generate the number of working hours corresponding to the credits assigned, which are furthermore to be controlled in an effective manner [7].

Bearing in mind this new frame in which we must develop our teaching activity, it is necessary to reconsider the role of mathematics in engineering schools. It is obvious that mathematics will continue fulfilling a double objective in the new university frame. On one hand they will continue being a powerful formative tool and, on the other hand, they will continue being the support and background of other academic disciplines. Therefore, a mathematical basic background will be basic in the new frame of "acquisition of competencies". And the methodological changes affect, in an essential way, the mathematical topics. Let's not forget that, at least in Spain, we will move from a eminently face-to-face teaching to a mixed teaching, in which the student, under the supervision of the instructor, must acquire new knowledge across his/her own activity. With this panorama, it is necessary to highlight the importance of e-learning and b-learning techniques in the new academic configuration.

There have been numerous experiences in this sense. Note should be made here of a precedent to the present project that we mentioned earlier. We are referring to the **Xmath** project [1,2,5], which has involved the design of a framework for online learning and a Calculus I course with a modular structure. Each module contains theory, exercises, tests, problems, miscellanea, puzzles, etc. and links to websites that are deemed to be of interest. A pilot course has been drawn up (Xmath Pilot Course). An assessment has been made of this course and the response by Norwegian, Slovak and Spanish students has been fairly homogenous [10]. Generally speaking, they consider the course to be interesting for reinforcing the subject and they appreciate the different levels of depth in the subject matter. The main criticisms are levelled at the design of the website and the links used in the course.

It is therefore necessary, even essential, to provide quality material that is flexible to use and that caters for both teachers and students when planning and seeking knowledge on an individual basis. This is the mission that underpins the project we shall present forthwith.

4 The dMath Project

Building a European Database of Mathematical e-learning modules is the aim of this new EU project. The main focus of dMath [3,4] is to use a new and, in part, untried technology within the web-based publication of technical and mathematical formulas. The project, which is being run under the supervision

of the Buskerud University College, Norway (HIBU), is a partnership between two commercial companies, Industriell Dokumentasjon a.s. (Norway) and Soft4Science (Germany) and five universities or colleges: Czech Technical University (Prague, Czech Republic), Sogndal College (Sogn og Fjordane, Norway), Slovak Technical University (Bratislava, Slovakia), Universidad Pontificia Comillas (Madrid, Spain) and Savonia Polytechnic (Kuopio, Finland). Salamanca University is the external evaluator of the project.

The dMath system (working title "SciLAS") contains:

1. A XML/MathML authoring tool (SciWriter).
2. A content management system (CMS) (working title "Orange").
3. A data repository (working title "dMathArchive") containing a number of independent modules (also known as Reusable Learning Objects).
4. Download facilities (using the SCORM package) to a local Learning Management System (LMS).

User friendliness are ensured by the interface and the fact that each module will be supported by illustrations and interactive animations in order to visualise the module content for the user. The mathematical calculations in the modules are performed via the Xmath Calculator.

The project is characterised by innovative pioneering work with respect to the use of technology and the way in which the database is made. The database will be an "evolutionary system" that is developed by the users, i.e. the users will be able to modify the existing modules and also create entirely new modules. The project group will therefore set up standards to ensure the uniformity of the different types of modules and the group will also act as "editor" of new and modified modules once the system is up and running.

dMath information can be found at: http://dmath.hibu.no/main.htm. The home page, can be shown in Figure 1. The different mathematical modules

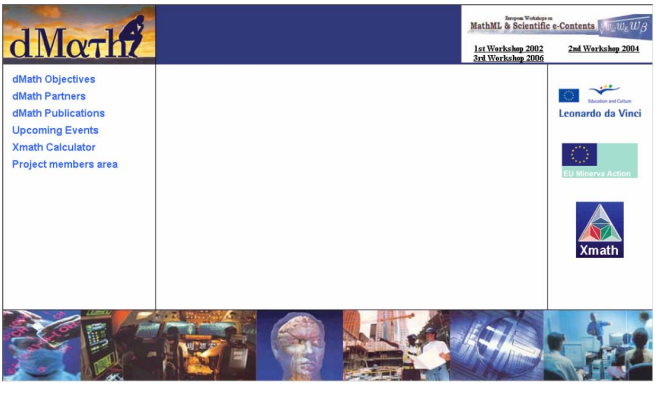


Fig. 1.

drawn up cover a large part of the content present in various mathematics subjects in Engineering Schools: Linear Algebra, Calculus, Differential Equations, Numerical Methods, Statistics, etc.. The possibility of being able to perform the calculations through the Xmath Calculator allows this material to be used in a flexible way, adapting it to the different needs of users.

5 About SciLAS

As mentioned earlier, part of the system created hinges upon the SciLAS database [8]. This is the brainchild of one of the project's partners (Soft4science, Germany) and it manages the entire database for the different modules created. It might be more straightforward to say that SciLAS is the result of combining the specifications of the dMath project with the CMS (Content Management System) and with SciWriter, which generates the .xml documents used to prepare each and every one of the project modules. The technology chosen is XML, which is to become the new standard in numerous facets: data format, data description, access to data, data transformation and data exchange.

The SciWriter processor, fine-tuned with the specifications provided by members of the dMath group, allows the creation of .xml files (and their export to the various market formats) in a more flexible manner, and has been used extensively for creating the different files that are part of the dMath project's various mathematical modules. Figure 2 shows part of one of the documents created, with the upper menu bars revealing the wide range of options provided by the use of the processor.

The database allows different levels of use. On one hand, there are the document creators (publishers) and, on the other, the users of the system. The handling of the database is extremely user-friendly and flexible and it has a tree storage structure; where the different topics, are separated into different units and in each one of them the Publisher can include one, or several, introductions to the subject, those theories as deemed appropriate, examples, applications, exercises and problems, and assessments. Furthermore, colour codes indicate the status of each one of the files that make up the different sections, both from the publishers' perspective, thereby enabling them to know whether or not the documents can be handled and modified, being opened or closed through a simple click of the mouse, as well as from the users', who at all times know which files are accessible and the overall structure of each unit. Moreover, the system automatically saves any modifications, so that at any given moment we know which version of the document we are working with.

The display of the different files for system users will allow these, on the basis of the files stored, to select the various documents they need: creation of mathematics courses with different levels, drafting of assessment material, preparation of material for individual tutorials, etc.

Users may export to SCORM those parts of the unit deemed to be more suitable for their purposes.

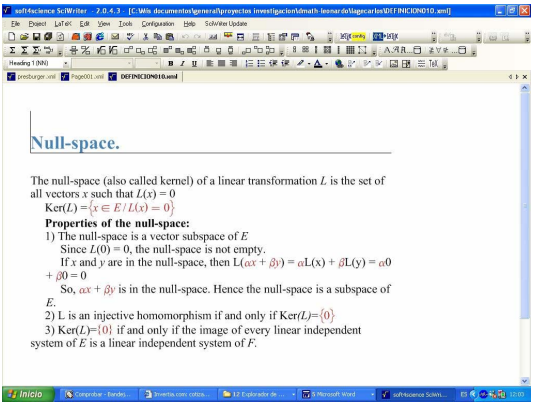


Fig. 2.

In short, SciLAS provides an extremely useful tool for all its potential users: educational organisations, companies, teachers, students, engineers and researchers in general. Its flexibility allows the preparation of customised documents, responding to the goals each user aims to achieve and providing teachers with a quick and efficient way of finding the material they need to use in the classroom. This means students receive the basic material they need in terms of their individual requirements, thereby contributing to their basic training.

Each user of the system can draft their own syllabus by means of the suitable combination of modules. Accordingly, the material placed at the disposal of the educational community constitutes a vast repository of mathematical resources that will provide a response to a broad range of learning situations.

6 Conclusions

The Bologna declaration has started a necessary revision of the contents and of the teaching techniques to use in the teaching of mathematics at the different University levels. In this sense, the new demands created by the necessary European harmonisation and the need to have available tools that may make possible a continuous mathematical formation in diverse learning situations, demand a new strategic response.

In order to begin to give an answer to these new challenges the dMath project has been developed and its main characteristics are introduced in this article.

Alter analysing the main characteristics and technical specifications of dMath we develop the innovative pioneering work of the project.

Acknowledgements

Gerardo Rodriguez's work has been partially financed by Junta de Castilla and Leon under the project US30/06.

References

1. Odd, B.: Xmath: Mathematical e-Learning Project in the EU Minerva Action. In: First European Workshop on MathML &Scientific e-contents. Abdus Salam International Centre for Theoretical Physics (ICTP), Trieste, Italy (November 2002), http://www4mail.org/mathml2002/index.html
2. Odd, B.: WebMathematica: Interactive Mathematics on the web. In: First European Workshop on MathML &Scientific e-contents. Abdus Salam International Centre for Theoretical Physics (ICTP), Trieste, Italy (November 2002), http://www4mail.org/mathml2002/index.html
3. Odd, B.: dMath project. Second European Workshop of MathML& Scientific e-Contents. Kuopio (Finland) (September 16-18, 2004), http://dmath.savonia-amk.fi/secondWorkshop/servlet/Home/program.html
4. Bringslid, O.: Building a European Database of Mathematical e-learning Modules. In: Proceedings of the IPSI-2005-Spain conference, Costa Brava Spain, 28-30 April (2005)
5. Canessa, E.: MathML technologies in the XMath e-Learning Project. In: First European Workshop on MathML &Scientific e-contents. Abdus Salam International Centre for Theoretical Physics (ICTP), Trieste, Italy (November 2002), http://www4mail.org/mathml2002/index.html
6. Directorate Generale for Education and Culture. In: ECTS Users' guide, Brussels (17 -08-2004)
7. García, A., García, F., Rodríguez, G., -Villa, G., de la, A.: Una propuesta de innovación educativa: Una ense ñanza integrada del cálculo infinitesimal en las escuelas de ingenier ía. In: Proceedings XIII Congreso de Innovación Educativa en las Enseñanzas Técnicas. Maspalomas (Gran Canaria), 22 y 23 September 2005, vol. 21 (2005)
8. Keil, Bernhard: Authoring MathML and e-Learning Content with SciWriter and SciLAS. In: Second European Workshop of MathML& Scientific e-Contents. Kuopio (Finland). 16-(September 18, 2004), http://dmath.savonia-amk.fi/secondWorkshop/servlet/Home/program.html
9. Las universidades y la enseñanza superior en el espacio europeo del conocimiento, Parlamento Europeo (September 2002)
10. Norstein, A., Gerardo, R., Velichova, D., de la Villa, A.: Students' impressions about XMath pilot course. In: Second European Workshop of MathML& Scientific e-Contents, Kuopio (Finland) (September 16-18, 2004), http://dmath.savonia-amk.fi/secondWorkshop/servlet/Home/program.html
11. Rodríguez, G.-V., de la, A.: Could the computers change the trends in Mathematics learning? A Spanish overview. In: Rodríguez, G. (eds.). Proceedings Applimat 2005. Slovak University of Technology Bratislava (Slovakia), (February 1-4 2005)
12. de la Villa, A: General Aspects about the Use of Computers in the Teaching. Advantages and Disadvantages. In: First European Workshop on MathML &Scientific e-contents. Abdus Salam International Centre for Theoretical Physics (ICTP), Trieste, Italy (November 2002), http://www4mail.org/mathml2002/index.html

A Study of Tool Support for the Evaluation of Programming Exercises

Heinz Dobler[1], Rudolf Ramler[2], and Klaus Wolfmaier[2]

[1] University of Applied Sciences, Softwarepark 11, 4232 Hagenberg, Austria
heinz.dobler@fh-hagenberg.at
[2] Software Competence Center Hagenberg, Softwarepark 21, 4232 Hagenberg, Austria
{rudolf.ramler,klaus.wolfmaier}@scch.at

Abstract. To foster the process of learning to program, theory and exercises are necessary. Traditionally, manual review of the source is used to provide feedback for the solutions. The effort is substantial and identified problems are prone to subjective interpretation. We study static analysis and testing tools as an enhancement to reviews and discuss the benefits. We analyze our findings by comparing the results from analysis by cross-checking the outcomes of the different tools with each other, with the tutors' remarks, and with the course outcome additionally taking into account final examination results. The effort was substantial and it turned out, that the tools are no replacement for manual review. Tool support is an enhancement due to localization of problem areas, accurate check of programming rules, and an efficient way to detect plagiarism.

1 Introduction

Courses in programming are the basis of most of the Computer Science and Software Engineering studies–especially of those following the paradigm "programming first" of the ACM Computing Curricula 2001 [1]. Teaching programming includes several topics such as algorithms and data structures, programming languages, libraries, frameworks, and development environments. To foster the difficult process of learning to program, both, theory and exercises are necessary. Exercises are best chosen from practical examples of substantial size [2]. Detailed and constructive feedback on the various solutions for these exercises is a vital part in learning how to program. Qualified feedback requires reviewing the source code and the documentation, analyzing and commenting errors and shortcomings, and giving hints for improvement. These tasks involve tutors, course assistants, and professors. At the University of Applied Sciences in Hagenberg, several courses follow this model of teaching.

Providing detailed feedback and hints for improvement for every individual student on a personal basis requires substantial effort and identified problems are–to a certain extent–prone to subjective interpretation. The aim of this work is to study the applicability of static analysis and testing tools [3] for evaluating solutions to exercises and to discuss the consequential benefit for tutors and course instructors.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 376–383, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Study Approach

Many tools based on a broad range of methods and techniques for analyzing the quality of software are available [3, 4, 5]. For this study, the following methods and techniques have been considered:

- **Manual Review** and **Inspection** are techniques that are widely applicable to a broad range of software development artifacts such as specification, design, source code, and documentation. Typically, these techniques help finding problems early in the development process which allows prompt feedback and fosters knowledge sharing and learning. Manual review and inspection spans from informal walk through over peer reviews to formal software inspection meetings [6].
- **Requirements-based Testing** and **Coverage Analysis** assures the quality of the program from two different perspectives: from the outside (black-box) and from the inside (white-box). Requirements-based testing [7] aligns test cases with available requirements and thus assures correct and complete implementation of the specification. Coverage analysis [7] aligns test cases with the code and, assures all elements such as statements or branches have been exercised. Thereby, on the one hand, coverage analysis supports finding new test cases to construct a more complete set of tests and, on the other hand, it identifies "dead" or unreachable code.
- **Static** and **Dynamic Analysis** reveals possible problem areas within a program taking into account only the source code or both, source code and dynamic behavior. Static analysis detects violations of programming guidelines, flaws in the program logic, harmful use of language features or standard libraries, and adverse design styles. It is usually based on a predefined set of analysis rules. Dynamic analysis considers the behavior of the analyzed program at run-time. It is typically targeted at detecting performance issues or memory leaks. Since dynamic analysis requires the execution of the program, results are only representative for the exercised scenario (test case).

A number of different exercises are handed out in the programming course in the third semester of the bachelor study Software Engineering. For the study presented in this paper, one of these exercises has been selected to apply the available tools on the students' solutions. The criteria used to select the exercise were: (1) it concerns a real-world problem the students are familiar with, (2) it is of adequate algorithmic difficulty, (3) the resulting program has a clearly specified interface, and (4) at least a dozen of complete and working solutions were available for the analysis.

The assignment of the selected exercise describes a module of an application in use in the university's course administration the students are familiar with. The exercise includes an algorithm to be implemented in C++ making best use of features provided by the standard template library [8]. To get unbiased results, neither the students nor the tutors and course instructors had been informed about this study in advance.

In order to automatically process and to compare the solutions provided by the students, some preparation work had to be done in advance:

- **Defining the sample:** From all available solutions, a subset has been selected that had to be amenable for further automated processing, i.e., the source code is compilable on the analysis platform as students were free to choose their favorite

development environment consisting of compiler and libraries on any operating system. Only 17 of the submitted solutions–about on third of the available solutions–fulfilled this criterion.

- **Sanitizing collected samples:** Furthermore, the submitted solutions needed some preparation so that all analysis steps could be completely automated. This preparation work included: (1) combining all source code fragments in one source file following a specified naming scheme, (2) adapting hard-coded file and directory names, (3) converting file formats, and (4) removing individual comment blocks to make the solutions anonymous.
- **Automating the environment:** The environment allowed automatically compiling the sources, deploying of binaries and test cases, executing the tests and static analysis runs, and collecting results. To be able to transfer and preserve the environment, it was set up as a virtual environment using Microsoft VirtualPC. Furthermore, platform specific tools such as Dr. Watson were configured to intercept program errors and system crashes.
- **Preparing manual review results:** The manual review results provided by tutors for each student solution are the main input to determine a student's course grade. The manual review results for the analyzed solutions plus the overall course grade of the students were collected and made comparable in an anonymous form.

The results of the automatic static and dynamic analysis techniques applied in this study where then compared to the manual review results. The goal of the study is an evaluation of the applicability and usability of the selected tools in the context of analyzing the students' solutions to exercises. In addition, the benefits and limitations of the selected quality assurance approaches are identified and discussed.

3 Evaluation Results of Studied Tools

Manual review is the typical technique applied by tutors. Based on the student's assignment submission, tutors typically perform a so-called desk check [6] for the provided source code. For the manual review to work effectively, following requirements must be met: (1) professors and course assistants must provide consistent review rules for the tutors, against which the reviews are conducted, (2) tutors must provide feedback to the students about the review results, and (3) professors, course assistants, and tutors must provide consultation-hours, so the students can discuss the review results. The results of the desk checks are the main input for a student's course grade. They are represented in the manual review results shown below and are used as basis in the comparison of the studied tools.

Automated tests were used to check the correctness and completeness of the solutions delivered by the students. A test framework had been developed in Python [9] to automatically run the programs on different input files and to compare their results with the expected ones. To be able to compare the test results of the different solutions, the test driver also "normalized" the program output as we did not want to change the source code of the students' solutions. A set of 54 different test cases was specified based on the assignment handed out as part of the exercise. Test design techniques such as equivalence class partitioning and border value analysis [7] were used. All test cases were verified against a sample solution. For each student solution

all test cases were executed and the results were classified in three categories: (1) *success*, if the program passed the test, (2) *failure*, if the program did not pass the test, i.e., the expected result did not mach the actual result, and (3) *error*, if the program crashed during test case execution. After executing all test cases for all solutions the results were combined in the test result matrix depicted in Fig. 1.

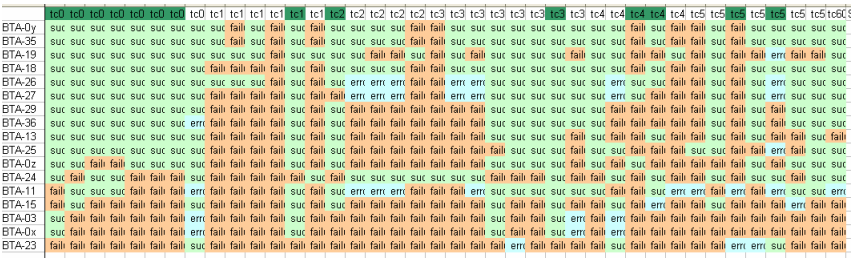


Fig. 1. Test Result Matrix

In combination with the test execution, a code coverage analysis was conducted. **Code coverage analysis** was used in two ways: First, in test case development a simultaneous coverage analysis helped to refine the set of test cases by identifying additional cases. Second, after a complete set of test cases had been defined and validated, the coverage results were used to evaluate the suitability and accurateness of the student's implementation in respect to the requirements specified in the exercise assignment. Uncovered code indicated either unreachable "dead" code or code that implemented additional functionality not specified and, thus, not reflected in the set of test cases. Dead code and additional, not required functionality have a negative impact on the program's maintainability and, besides, bears the risk of harmful unexpected behavior and security leaks. *BullseyeCoverage* [10] was used to measure the test coverage. The tool provides two different types of coverage measurement techniques: function coverage and condition/decision coverage. Figure 2 shows an example coverage report.

Static program analysis [11] was used to evaluate the code quality of the solutions provided by the students. The objective of static analysis is to reveal potential defects or to indicate defect-prone parts in the source code or a related document [7]. For static analysis the popular tool *pclint* [12] was used. The tool *pclint* by Gimpel Software is a software package that finds bugs, glitches, inconsistencies, non-portable constructs, and other sources of problems in C/C++ programs. A typical rule, taken from *pclint*, might look like this: "*destructor for base class 'Symbol' is not virtual … It is conventional for inherited classes to have virtual destructor so that it is safe to delete a pointer to a base class"* [12]. The available static analysis rules are combined to different groups representing different severity levels. This allows defining the quality level in a project but needs additional customization of the rule set. The rule set applied in the study included all messages of level *Error*, *Warning*, *Info*, and *Note*. The analysis with *pclint* revealed between 36 and 132 messages with an average of 82 messages for the different solutions (Figure 3). About 10 percent of the messages have been classified as errors or warnings.

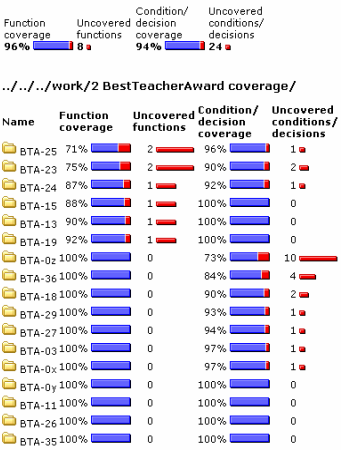


Fig. 2. Test Coverage Results

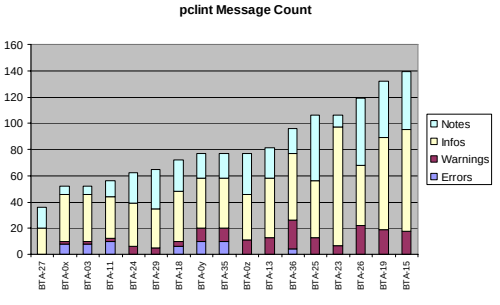


Fig. 3. pclint Message Count

Finally, the enterprise code analysis tool embedded in *Microsoft Visual Studio 2005* [13] has been used to detect programming-language specific issues in the solutions delivered by the students. *Microsoft Visual Studio 2005* provides several **compiler warnings** to identify problems such as syntax checking, checking for correct data type usage, and checking interface consistency. In addition, the *Microsoft Visual Studio 2005* compiler includes an option that also allows checking for "advanced" rules, such as the usage of uninitialized memory, invalid pointer dereferenciation or memory leaks by **symbolic execution** [14]. It is a complementary approach to other static analysis tools as it only identifies "real" defects within the code and omits warnings about potential problems. The analysis with the Microsoft Visual Studio 2005 compiler revealed 0 to 6 messages (avg. of 2.3 messages) for the different solutions.

4 Discussion of Findings

Manual review results were taken from the evaluation done by the tutors during the programming course. These results are verified by the course assistants and are

incorporated into the course grade. No further manual reviews were conducted as part of this study. The disadvantage of manual reviews is the high effort for tutors, course assistants, and professors. Manual reviews are conducted for all student solutions on review guidelines provided by the course assistants and professors. Feedback and suggestions for improvements are given on an individual basis in form of written comments and in consultation hours. Furthermore, since the reviews are typically conducted by several tutors working in parallel, findings are prone to subjective interpretation and need additional cross-checking and verification.

We found that **automated tests** are highly useful to assess the correctness and completeness of the students' solutions for the exercise. The manual investigation of test results showed that tests failed because of errors (e.g., illegal pointer arithmetic) and requirements that were not implemented (e.g., missing input validation). When the tests passed, the requirements specified in the exercise assignment had been implemented correctly. However, other aspects such as programming style and design might still contain problems. The effort involved in providing an environment for automated test execution, in designing test cases, and in analyzing test results is high. To be able to use automated tests in order to measure the quality of a solution it is necessary that the set of test cases is complete, i.e., all requirements are covered by a representative set of test cases, and the test results are analyzed manually to identify defects. The number of failed or passed test cases is not a suitable measure of quality. Yet, the number of failed test cases may be an indicator for further investigation.

Analysis showed that only 4 solutions had full **function coverage** and **condition/decision coverage**. The main reason for the lack of coverage was that student solutions were not in alignment with the exercise assignment. The solutions contained about 15 to 20 percent additional code not necessary for solving the exercise, e.g., due to over-generalization. Some solutions contained also dead code, i.e., code that cannot be reached in any execution scenario. The overall effort for conducting the coverage analysis was low. However, the precondition for a coverage analysis is an adequate set of test cases. Furthermore, a manual investigation for interpreting the results is still required. For example, to get a first overview, we used function coverage to assess the alignment of a student solution and condition/decision coverage to assess the structural quality, e.g., to detect dead code.

Static program analysis allows an automatically checking of low-level programming guidelines, a monotonous task for human reviewers. Tutors can use the saved time to focus on advanced aspects of the student solutions that cannot be checked automatically, such as design or algorithms. While the initial effort for conducting a static analysis seems low, deriving useful insights requires additional effort. We found that using *pclint* produces a huge number of often irrelevant messages. Therefore, it is necessary to refine the rule set by limiting the messages to certain severity levels such as *Error* or *Warning* and by calculating the number of violated rules ("distinct count") instead of taking the absolute number of warnings. Furthermore, we manually investigated the messages from the static analysis to verify that the tools actually indicate defects. All of the investigated messages revealed low-level defects within the code. However, the line numbers specified as part of the messages were usually not the position of the actual defect in the code. Thus, verifying the messages required "manual debugging", which needed a lot of experience and time.

Modern compilers such as the one included in the Microsoft Visual Studio 2005 development environment provide **compiler warnings** and support checking of advanced rules by **symbolic execution** [15] and are thus able to detect memory leaks or usage of uninitialized variables in addition to syntax errors. No compiling errors were found because these checks were part of the entry criteria for selecting a student solution. The check of advanced rules by symbolic execution revealed up to 6 errors per solution, where each of the errors indicated a serious problem. The effort for conducting the analysis was low as the tool was integrated in the development environment and no further processing of the messages was done. However, the number of relevant messages and thus the insights provided were small, too.

Figure 4 compares the **results of the different tools** for the different student solutions side by side. The different tools evaluate the student solutions from different points of view. To visualize the results of the different tools in a single figure, we therefore normalized the results from the tools according to an absolute threshold individually defined for each tool. Consequently, the threshold indicates the acceptable number of issues raised for a particular viewpoint, e.g., the number of distinct *pclint* messages or the minimum coverage required.

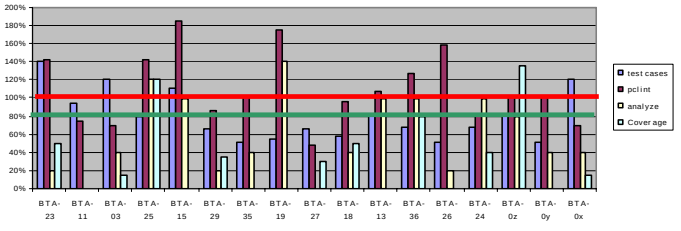


Fig. 4. Synopsis of Analysis Results

5 Conclusion

We found that the different approaches and tools applied cover the quality of the solutions from very different points of view such as the conceptual solution design, the correct implementation of requirements, the compliance to programming rules, and the syntactical correctness.

Each of the tools had its strengths and was able to point out relevant quality issues. However, a positive correlation between the results of one of the tools, tutor reviews and final marks could not be supported by the study. From these findings we conclude that none of the tools is suited to replace the current practice of manual reviews by tutors and instructors. Tools are a valuable enhancement to manual reviews. The results delivered by the different tools support the localization of problem areas and an accurate checking of low-level programming issues. This helps to redirect costly tutor work from routine work such as analyzing program code, checking rules, etc. to focus more on the objectives of the exercise, the algorithms and data structures, and supporting students in the role of a coach on an individual basis. The benefits of the tools have to be weighted against the often high effort involved in applying the tools effectively. In many cases, a tool-supported analysis requires setting up and maintaining an

automation infrastructure, precisely specifying administrative requirements to process the solutions, manual investigation and interpretation of results, as well as the acquisition of tool-specific know-how. These high initial costs only pay off when tools are consequently used over several exercises or courses.

The tools have been used by the researchers from the point of view of instructors and tutors. Offering these tools to students already for developing their solutions could be a good strategy to increase their learning success. The use of "objective" measures applied in advance of tutor feedback helps students to get acquainted with common development and quality assurance tools, to think for themselves and seems suitable to improve their quality awareness. Software engineering students in Hagenberg usually start using these tools in courses, where a real-life project is conducted. Introducing the practices mentioned above already within basic programming courses, however, demands additional work by tutors and bears the risk of distracting students from the main goal of the course. So for basic programming courses the authors propose to continue with the traditional system of evaluating student exercises using tutors but encourage students to improve their results by using tools before delivering.

Acknowledgement. The authors gratefully acknowledge support by the Austrian Government, the State of Upper Austria, and the Johannes Kepler University Linz in the framework of the Kplus Competence Center Program.

References

[1] The Joint Task Force: Computing Curricula 2001, Computer Science, ACM (2001)
[2] Jazayeri, M.: The Education of a Software Engineer. In: Proceedings of 19th IEEE Conference on Automated Software Engineering, Linz, Austria (2004)
[3] Ghezzi, C., Jazayeri, M., Mandrioli, D.: Fundamentals of Software Engineering, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)
[4] Tian, J.: Software Quality Engineering – Testing, Quality Assurance, and Quantifiable Improvement. John Wiley & Sons, West Sussex, England (2005)
[5] McConnel, S.: Code Complete. Microsoft Press, Redmond, Washington (2004)
[6] Wiegers, K.E.: Peer Reviews in Software. Addison Wesley, London, UK (2002)
[7] Spillner, A., Linz, T., Schaefer, H.: Software Testing Foundations, dpunkt (2006)
[8] Josuttis, N.: The C++ Standard Library–A Tutorial and Reference. Addison-Wesley, London, UK (1999)
[9] The Python Programming Language, http://www.python.org
[10] Bullseye Code Coverage Tool, http://www.bullseye.com
[11] Ramler, R., Wolfmaier, K., Dobler, H., Altmann, J.: State of the Art in Static Program Analysis and Unit Testing (in German), Technical Report SCCH 0323 (2003)
[12] pclint Static Analysis Tool, http://www.gimpel.com
[13] Visual Studio, Development Environment (2005), http://www.microsoft.com
[14] Bush, W.R., Pincus, J.D., Sielaff, D.J.: A static analyzer for finding dynamic programming errors. Software—Practice and Experience (June 2000)
[15] Larus, J.R., Ball, T., Das, M., DeLine, R., Fähndrich, M., Pincus, J., Rajamani, S.K., Venkatapathy, R.: Righting Software. IEEE Software (May/June 2004)

Aplying a Semantic Hypermedia Model to Adaptive Concept Maps in Education

Fernando Molina-Ortiz[1], Nuria Medina-Medina[2], and Lina García-Cabrera[3]

[1,2] University of Granada. E.T.S.I.I.T. 18071, Granada, Spain
{fmo, nmedina}@ugr.es
[3]University of Jaén. Campus Las Lagunillas. 23071, Jaén, Spain
lina@ujaen.es

Abstract. We present a model and a tool that allow the development of concept maps that can be navigated by the student, thereby allowing plenty of course contents to be browsed through a concept map. The tool applies several user adaptation techniques that reduce disorientation and cognitive overhead when browsing large maps by adapting the concept maps to the student's knowledge. In addition, an evolutionary framework is used so consistency after changes is always guaranteed. The paper also describes a preliminary application to a real case at a University course.

1 Introduction

Concept Maps are diagrams that represent organized knowledge, showing concepts and the relations between them [1]. They have been shown to be very useful in education [2], and we believe that the utilization of Information Technology and Computer Science to support the use of concept maps further enhances the educational experience.

For us, a navigable concept map is a concept map that is able to present additional information when the user selects one of its elements. In this way, if the user is interested in a concept, the system allows to visualize one or more electronic documents or web pages that permits the user to learn about the concept. In an educational context, this allows to present to the user (the student in this case) plenty of course contents, which are given structure and coherence through a concept map. Therefore, the student can explore (navigate) the educational contents in a non-linear way, choosing first the elements associated to the concepts in which he is most interested, and following the strategy he prefers.

Nevertheless, this freedom in exploring the course contents can lead to problems. The student can feel overwhelmed by a big or unknown concept map, which may show too many elements at once or provide material that the student is not yet prepared to understand, leading to a helpless and demotivated student. In order to reduce these problems we apply techniques from the field of adaptive hypermedia systems [3], so we can provide concept maps that are adjusted to the student's features and knowledge, offering only the information that the student is ready to comprehend.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 384–391, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In this paper we present a model, SEM-HP [4,5], and an author and browsing tool, JSEM-HP [6]. They have their origin in the field of hypermedia systems, and provide a formal model and an environment for the development of adaptive hypermedia systems based on navigable concept maps. Section 2 introduces SEM-HP and JSEM-HP, and section 3 describes a preliminary case example. Finally, related work, conclusions and further work are discussed in section 4.

2 The Model SEM-HP and the Tool JSEM-HP

SEM-HP [4,5] is a Systemic, Semantic and Evolutionary model that can be used for the development of educational systems, in which concept maps are key elements and that supports user adaptation. JSEM-HP [6] is a tool that implements the main elements of this model.

A SEM-HP system is composed by four interacting subsystems: memorization, presentation, navigation and learning. The memorization subsystem stores, structures, and maintains the contents that will be offered to the students in the form of a global navigable conceptual map. The presentation subsystem allows the author to select subsets of a conceptual map in order to create more focused and simpler conceptual maps. The navigation subsystem provides tools to decide the order in which the information can be browsed, and the learning subsystem takes care of the user modeling and user adaptation.

SEM-HP also supports the evolution of the systems created with it since in every step of their maintenance it is guaranteed that the system being modified is consistent. This is performed using three main mechanisms: evolutionary actions, restrictions and change propagation. The evolutionary actions are initiated by the author of the system in order to change it, and are only executed if a set of restrictions holds, which in turn assures that the system will not get to an unusable state. The change propagation mechanism adjusts the whole system to a change, in order to keep it globally consistent: for example, if a concept is removed in the memorization subsystem, the change is propagated into the memorization subsystem itself by removing the associations to that concept, and into the other subsystems by also removing the concept and other elements that reference it. For that taking care of evolution we define a meta level, called the Metasystem, which receives changes from the author and applies them properly or rejects them if they can not be performed.

In the following sections we describe with more detail the main elements of the SEM-HP model as they have been implemented in the JSEM-HP tool.

2.1 Memorization Subsystem

The Memorization Subsystem stores, structures, and maintains the contents that will be offered to the students in the form of a global conceptual map that is modeled as a conceptual structure of memorization (CSm).

The CSm is a semantic network that integrates the conceptual map with the information offered to the students, having two kinds of nodes: concepts and items,

and two kinds of associations: conceptual and functional associations. Conceptual maps are represented with concepts and with conceptual associations, which connect concepts and have an associated label, such as 'part of', 'is a', etc. There is a special concept, called the root concept, which is the central element of the CSm. Items represent the pieces of information offered to the student, and can be a text document, an image, a video, etc. They are connected to concepts through functional associations, which define the role the item has in defining the concept (i.e. definition, example, explanation, etc). Figure 1 shows an example of CSm being edited with JSEM-HP, whose root concept is 'OO Concepts'. The main window shows the CSm, in which concepts are represented with ovals, items with rectangles, conceptual relations with arrows and functional associations with lines. The figure also shows two of the windows that can be opened to edit the elements in the CSm. The first one is for editing an item, which has, among other attributes, one URL which points to the document represented by the item. The smallest one shows a list that presents the available functional associations, which can be modified by the author to add more types of associations.

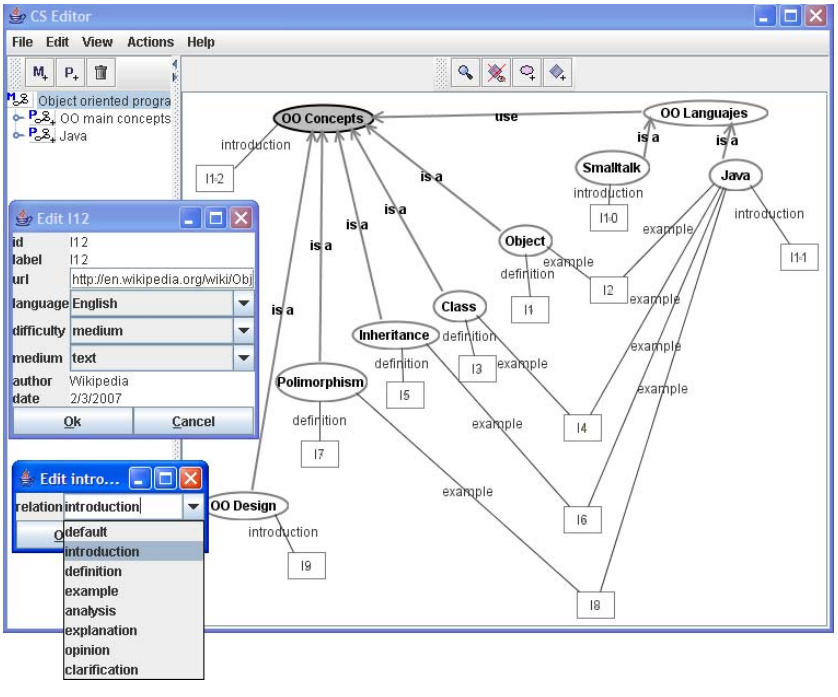


Fig. 1. Example conceptual structure of memorization

2.2 Presentation Subsystem

The Presentation Subsystem allows the author to select subsets of a conceptual structure of memorization in order to create more focused and simpler conceptual

maps, which are called conceptual structures of presentation (CSp). The subset is created by hiding or showing elements in the CSm in which it is based. In addition, this subsystem allows the author to change the presentation by modifying the layout of the original CSm, so it suits better to the new CSp. Figure 2 shows two possible presentations that can he defined based on the CSm in figure 1.

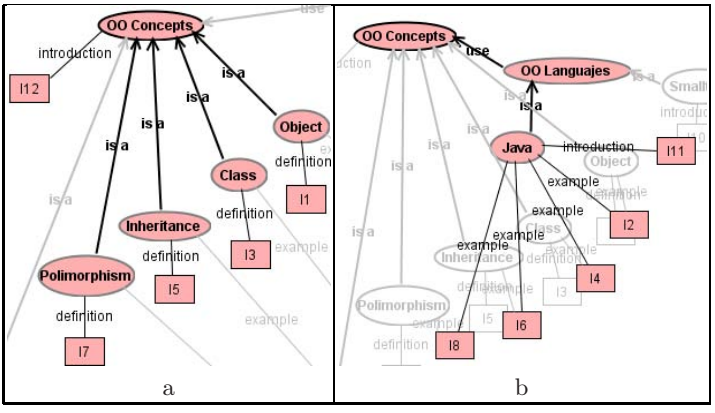


Fig. 2. Two possible presentations

2.3 Learning Subsystem

The Learning Subsystem is in charge of user adaptation. Its main elements are the user model and knowledge and update rules. A set of update and knowledge rules associated to a CSp define a conceptual structure of learning (CSl).

The user model stores the user features that will be used for adaptation. Among other information, it keeps the user's knowledge degree about the items offered by the system. This is represented with one of these semantic labels: null, low, medium, high and total, each of which has an associated numerical value ranging from 0 to 4.

Knowledge rules define the minimum knowledge that the student needs in order to be ready to understand an item. For example, figure 3 shows two possible knowledge rules for the item I5 in figure 2a, each of which defines one way in which the student can get ready to understand I5 (by having at least a medium knowledge about I12, or about I1 and I3).

Update rules model how the student's knowledge increases while he navigates the conceptual map. Each item has an associated update rule, which has a set of update predicates that are fired when the user visits the item. Each predicate defines a knowledge update about an item, so the author can define a knowledge update about the visited item, but also about other related items. The update of the user's knowledge about an item can be incremental (the user's knowledge is increased), or fixed (it is set to a given degree); absolute (the predicate uses a specific semantic label) or relative (it is based on the degree of user knowledge about the visited item); and first-time (performed only first time the item in the

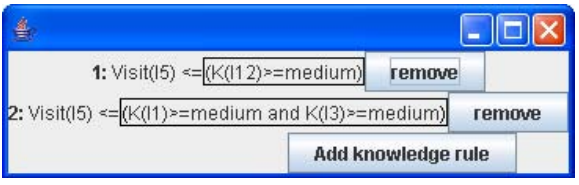


Fig. 3. Example Knowledge rules

head of the rule is visited) or each-time (it is run every time that the head item is visited). Figure 4 shows a possible update rule for the item I12 in figure 2a, which increases the student's knowledge about I12 in two levels every time it is visited, as well as setting the student's knowledge about I1 and I3 to medium, since the author considers that by reading the introduction to object oriented programming concepts (I12), the student also gains medium knowledge about the definitions of object and class (I1 and I3).

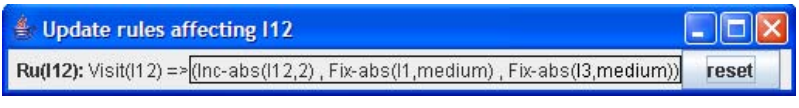


Fig. 4. Example update rule

2.4 Navigation by the User

Figure 5 shows a browsing session using the conceptual structure of learning defined in figures 2a, 3 and 4. The left window shows the conceptual map that is being navigated, the right window is a web browser that shows the currently selected item (I12) and the lower window, which is normally hidden, shows the current student's knowledge state (after visiting I12 twice). In the conceptual map, the items that the student is not ready to understand because his knowledge state does not satisfy any of its knowledge rules are disabled and partially hidden (this is the case of I7). In addition, the concepts and items are annotated with a darker color as the student's knowledge about them increases, so he is aware of his learning process.

3 Case Example

We have started to use JSEM-HP to aid in our teaching of the subject "Object Oriented Programing" that is taught (in Spanish) at the University of Granada. Different concept maps are used in the initial lectures to introduce the main concepts, and in the subject's web site in order to ease the navigation through the web's contents. These maps were initially created and exported to html with Microsoft Word, which caused problems with non-Microsoft's web browsers and

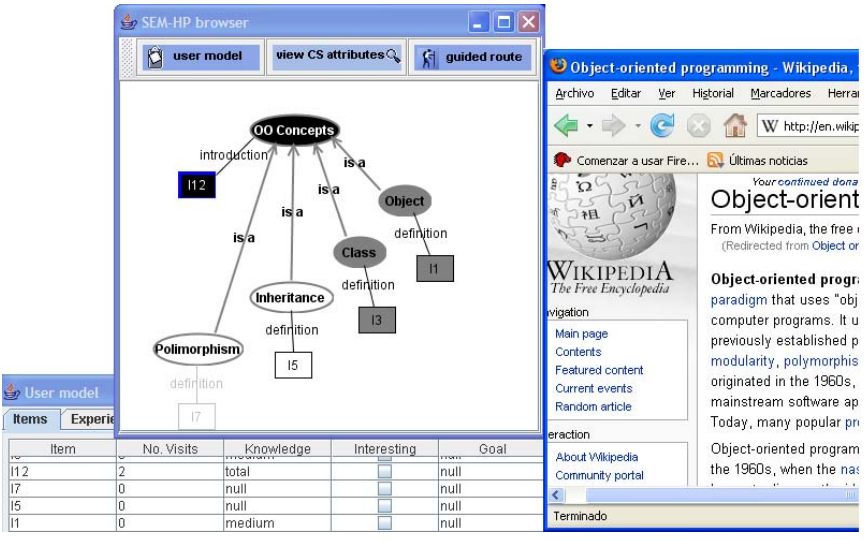


Fig. 5. Navigation by the user

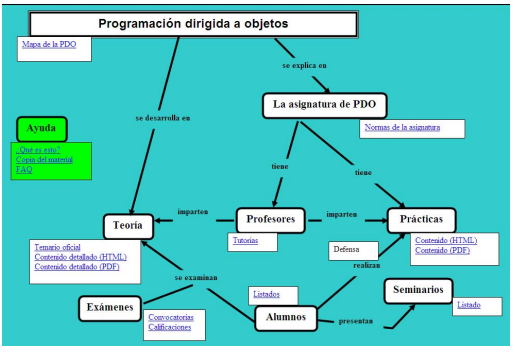


Fig. 6. Initial concept map done with Microsoft Word

made changes to existing maps difficult. JSEM-HP does not have these problems, so we decided to use the tool to redefine these concept maps, so at a first stage we could test SEM-HP's capabilities in knowledge representation.

Figure 6 shows the initial concept map (in Spanish), and figure 7 shows SEM-HP's version, translated to English. We can see that they are equivalent, aside from: a) the concepts Students, Exams and Theory are associated in JSEM-HP with three binary associations, while in the original map they are joined by a single tertiary association (meaning that students take theory exams), because JSEM-HP does not support tertiary associations; and b) the original map shows an isolated concept that is converted to the Help concept in the new map, which

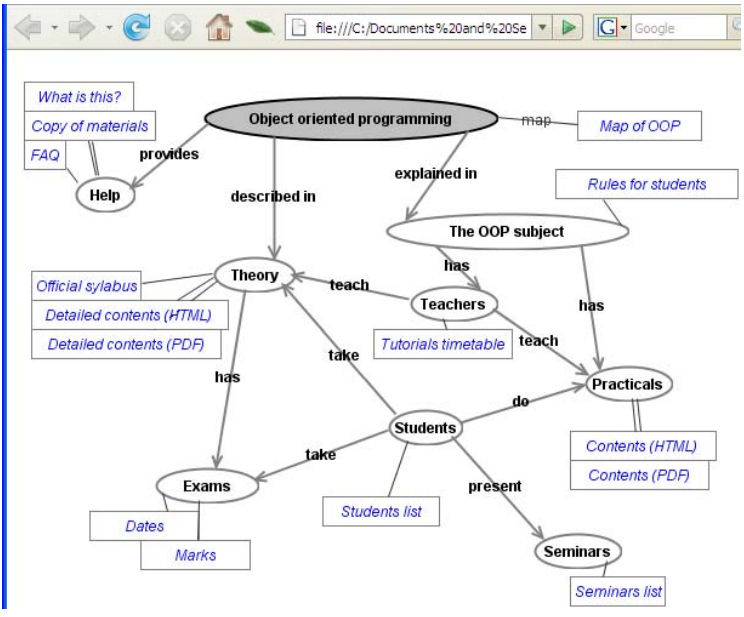


Fig. 7. Current concept map done with JSEM-HP

is connected to the root Object Oriented Programming concept, because SEM-HP does not allow unconnected concepts in order to keep consistency.

The result of this experience was that the original author preferred SEM-HP's version, since the lack of tertiary associations could be solved well for him with binary associations, and the ease of changing the already created maps and JSEM-HP's compatibility with any browser were important benefits for him. Being forced by the tool to connect all the concepts was not considered intrusive (in fact it allowed to create a more consistent concept map), and the explicit definition of the root concept required by JSEM-HP was straightforward for the author. In addition, the author positively valued the possibility of defining knowledge and update rules and using JSEM-HP's adaptive features.

4 Related Work, Conclusions and Further Work

There are two areas in which we can find different related works: conceptual maps and adaptive hypermedia systems. Regarding tools for the development of conceptual maps, we would like to mention CmapTools [7], which implements navigable concept maps that can also be shared and commented. CmapTools is more mature than JSEM-HP, and the main difference lies in its lack of user adaptation and evolutionary support. Regarding adaptive hypermedia systems, we can mention, among other relevant work, the AHA! architecture [8], based on the AHAM model [9], which employs several adaptive techniques that are similar to SEM-HP's, although they are focused to a more traditional web browsing

instead of concept maps. None of the reviewed works, in both fields, employ SEM-HP's evolutionary approach, that guarantees the system's consistency during its development and evolution. To this extent, we believe that SEM-HP's main contribution is grouping its semantic orientation by means of navigable concept maps with its user adaptation and evolutionary capabilities.

Our further work includes expanding the SEM-HP model to allow new elements such as tertiary associations, refining and expanding JSEM-HP to include more features already defined in the model such as feedback [10], and using the tool in more real cases in order to validate its usefulness.

References

1. Novak, J.D., Gowin, D.B.: Learning how to learn. Cambridge University Press, Cambridge (1984)
2. Horton, P.B., McConney, A.A., Gallo, M., Woods, A.L., Senn, G.J., Hamelin, D.: An investigation of the effectiveness of concept mapping as an instructional tool. Science Education 77(1), 95–111 (1993)
3. Brusilovsky, P.: Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction 6(2-3), 87–129 (1996)
4. Garcáa-Cabrera, L., Rodríguez-Fórtiz, M.J., Parets-Llorca, J.: Evolving hypermedia systems: a layered software architecture. Journal of Software Maintenance and Evolution: Research and Practice 14(5), 389–405 (2002)
5. Medina-Medina, N., Molina-Ortiz, F., Garcia-Cabrera, L.: Diversity of structures and adaptive methods on an evolutionary hypermedia system. IEE Proceedings - Software 152(3), 119–126 (2005)
6. Molina-Ortiz, F., Medina-Medina, N., García-Cabrera, L.: An author tool based on SEM-HP for the creation and evolution of adaptive hypermedia systems. In: ICWE '06: Workshop proceedings of the sixth international conference on Web engineering, ACM Press, New York (2006)
7. Cañas, A.J., Carff, R., Hill, G., Carvalho, M.M., Arguedas, M., Eskridge, T.C., Lott, J., Carvajal, R.: Concept maps: Integrating knowledge and information visualization. In: Tergan, S.-O., Keller, T. (eds.) Knowledge and Information Visualization. LNCS, vol. 3426, pp. 205–219. Springer, Heidelberg (2005)
8. De Bra, P., Aerts, A., Berden, B., Lange, B.D., Rousseau, B., Santic, T., Smits, D., Stash, N.: Aha! the adaptive hypermedia architecture. In: Proceedings of the ACM Hypertext Conference, Ottingham, UK, ACM Press, New York (2003)
9. Wu, H., Houben, G., Bra, P.D.: Aham: A reference model to support adaptive hypermedia authoring. In: Proceedings of the Conference on Information Science, Antwerp, pp. 51–76 (1998)
10. Medina-Medina, N., Molina-Ortiz, F., García-Cabrera, L.: An adaptation method by feedback in an evolutionary hypermedia system. In: ICWE '06: Proceedings of the 6th international conference on Web engineering, pp. 161–168. ACM Press, New York (2006)

Towards Virtual Course Evaluation Using Web Intelligence

M.E. Zorrilla[1], D. Marín[1], and E. Álvarez[2]

[1] Department of Mathematics, Statistics and Computation, University of Cantabria.
Avda. de los Castros s/n 39005 Santander, Spain
{zorrillm,david.marin}@unican.es
[2] Department of Applied Mathematics and Computer Science, University of Cantabria.
Avda. de los Castros s/n 39005 Santander, Spain
alvareze@unican.es

Abstract. Web-based learning environments are now extensively used. To guarantee the success in the learning processes, instructors require tools which help them to understand how these systems are used by their students, so that they can undertake more informed actions. Therefore, the aim of this paper is to show a Monitoring and Analysis Tool for E-learning Platforms (MATEP) which is being developed in the University of Cantabria (UC) to help instructors in these tasks. For this, web intelligence techniques are used.

1 Introduction and Motivation

In the last decade, the World Wide Web and new technologies have changed the concept of teaching. E-learning platforms are no longer expected to continue being data warehouses, but they are becoming a central element to the learning process [2].

The success of E-learning platforms such as WebCT or Moodle, is based on providing easy to use tools and offering students and teachers the possibility to connect and work "any-time, any-where". Nevertheless, these systems present certain deficiencies as they lack a face-to-face student-teacher relationship, which is manifested in facts such as: teachers do not really control the evolution of their students, and students cannot express their problems in a natural way.

Although some platforms offer reporting tools, the information which they provide is not enough to analyze the behaviour and evolution of each student. Futhermore, when the number of students and the diversity of interactions are high, the instructor has serious difficulties extracting useful information. Here is where web intelligence techniques play their role, given that these can generate from data statistics, analytic models and uncover meaningful patterns.

Currently, there are many general tools on the market with which develop solutions for making decisions (SQL Server, Business Objects, Pentaho, …). However, there are no specific tools to monitor, understand and assess the distance learning process of the students, although some effort is being made by universities.

CourseVis [3] is a tool that takes student tracking data collected by Content Management Systems and generates graphical representations that can be used by

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 392–399, 2007.
© Springer-Verlag Berlin Heidelberg 2007

instructors to gain an understanding of what is happening in distance learning classes. It directly uses web log files. It neither builds web sessions nor obtains behaviour patterns. Collaborative Analysis Tool (ColAT) [1] is a tool that offers interpretative views of the activity developed by students in a collaborative environment. It integrates the information of user actions from log files with contextual information (events, actions and activities) in order to reconstruct the learning process. It only shows statistic information. Mostow et al. in [6] describe a tool that shows a hierarchical representation of tutor-student interaction taken from log files.

On the other hand, in the data mining field, we find particular solutions to specific goals. For example, Zaïane [9] suggests the use of web mining techniques to build an agent that could recommend on-line learning activities or shortcuts in a course web site based on learners' access history. Tang et al. [8] are working in a recommender system that finds relevant content on the web and personalizes and adapts this content based on the system's observation of the learners and the accumulated ratings given by the learners. Finally, it is worth mentioning TADA-ED [4], a tool which tries to integrate various visualization and data mining facilities to help teachers in the pedagogical discovering process.

Our tool tries to join and integrate both perspectives: to offer instructors the useful information in static and OLAP reports and to show them the discovered student behaviour patterns.

This paper presents the Monitoring and Analysis Tool for E-learning Platforms (MATEP) that is being developed to help instructors to understand how these environments are used by their students, so that they can consequently make better decisions. This would allow them to re-structure the course according to its use, design the course activities according to the resources they have used, propose activities which encourage students to follow the course regularly and so on.

The paper is organised as follows. In section 2, we provide a general description of the Web Using Mining project inside which MATEP has been developed. In section 3, we specify the requirements demanded of the tool and show some reports obtained with it. In section 4, we give some recommendations based on the lessons learned. Finally, in section 5, we close with the conclusions drawn and the future work lines.

2 Web Using Mining Project

This project [10] initially started as an attempt to give specific answers to professors who, committed to these new methods of learning based on new technologies, do not get the appropriate feedback compared to the feedback they get from their students with traditional teaching methods. What we finally did, though, was to propose and design a global solution that also offered answers to the rest of people involved in the teaching environment, such as students, academic people in charge and site administrators.

The project is being developed following the stages of a web using mining project [7], although some stages have been added, such as the building of a data web house adapted to an e-learning environment [11], the generation of OLAP cubes [10], the developing of a reporting tool (MATEP) and the proposal to design, in the near future, a recommender agent. The architecture of the whole solution can be seen in [5].

3 Monitoring and Analysis Tool for E-Learning Platforms

3.1 Requirements

Instructors require a usable and specific tool that allow them track online the learners' activities and give them answers to general questions such as:

- Regards the course follow-up: When do students connect to the system? Do they work online? Could the value of a session be measured in relation to learning objectives? This would help teachers to carry out continual evaluation.
- Regards the course: How often do they use collaborative tools? What are the sequences of visited pages in each session, in which order, and how long do students stay in each one? What are the most frequent paths? This will allow teachers to discover if students follow the sequence established by them, to detect not-visited pages or to know which resources actually prefer the students (pdf, videotutorials, etc) among other things. In this way, teachers will have information to modify the structure of their courses and to adapt them to learners' behaviour.
- Regards students: What are the students' profiles? Is there a relationship between their behaviour and their qualifications? Who leaves the course and when?

In order for the application to be generally well accepted, the following requirements have been established:

- Web application. The tool must be accessible through Internet.
- Usable. It must present a clear and simple interface to find the information. Futhermore, this must not be more than two or three clicks.
- Easy to interpret. The tool must include expressive and intuitive reports and graphics with the goal that instructors understand the information at a glance.
- Interactive. Whenever possible, the tool must allow the professor's interaction. (dynamic reports).
- "Online" feedback. The webhouse must be fed every day from the e-learning platform in order for educators to have the information updated.

3.2 MATEP Tool

Figure 1 shows the MATEP welcome page. Once the instructor identifies himself, he must select the folder associated to his interest. Reports are initially organized in three folders: course follow-up, course usage and student tracking. When he selects a folder, a new page with the available reports is shown (see Figure 2).

Next, we present some of these reports. As the test data, we have selected a distance multimedia course of the UC which was held in the Spring semester 2006. We have the following information stored: for each session (obtained from the web log), we have the number of requested pages, time spent on each one and on the whole session, the date and time. Additionally, we have demographic information such as age and gender; academic information such as the degree course they are taking (background knowledge), partial and final marks and whether this is the first time they have selected this subject; and finally, course information such as page classification, estimated time to read and study the contents of each page and the course planning

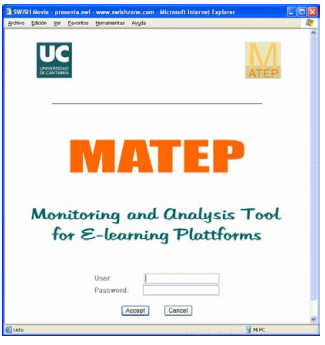


Fig. 1. MATEP welcome page

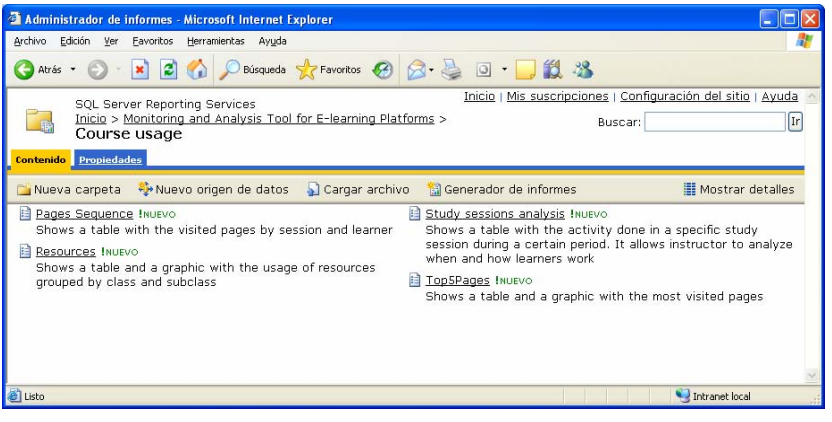


Fig. 2. Reports in Course Usage folder

(submission of home-based tasks, exams,…) and organization (definition of study sessions, pages to visit in each study session, ...).

Figure 3 shows two reports which summarize the global usage analysis of the course (Fig. 4a) and the usage per learner (Fig. 4b). Both include the number of sessions, the average time per session and the number of pages per session and, additionally, the second one also presents the average values of the course, so the instructor can compare the figures. These reports are very useful because they allow teachers to evaluate the usage of their course, to detect whether a student is about to drop out of the course, if the students connect to the system frequently, if the effort is greater than they planned, etc.

As can be observed, the reports have parameters with which instructors can modify the query and visualize the information in an aggregated or more detailed way.

Other interesting information for teachers when they have to design activities is to know how the students use the resources. To answer this question, the report in Figure 4 has been designed. The class and subclass attributes are established by instructors for each course.

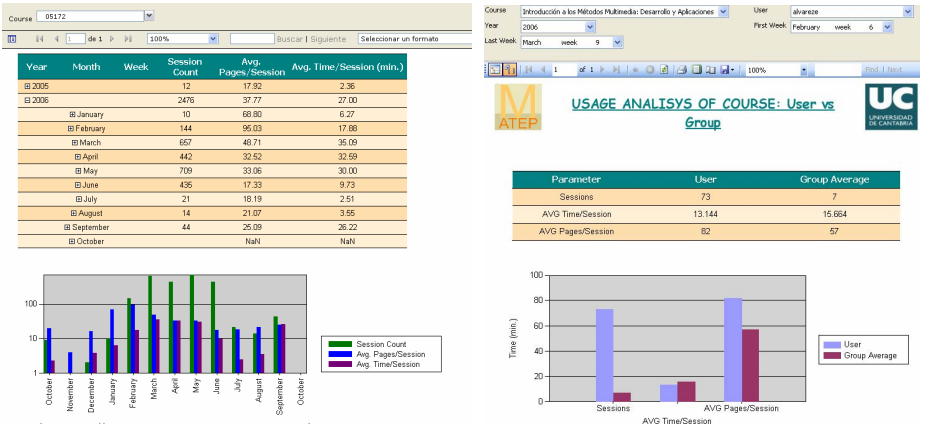


Fig. 3. Global course usage and usage per learner

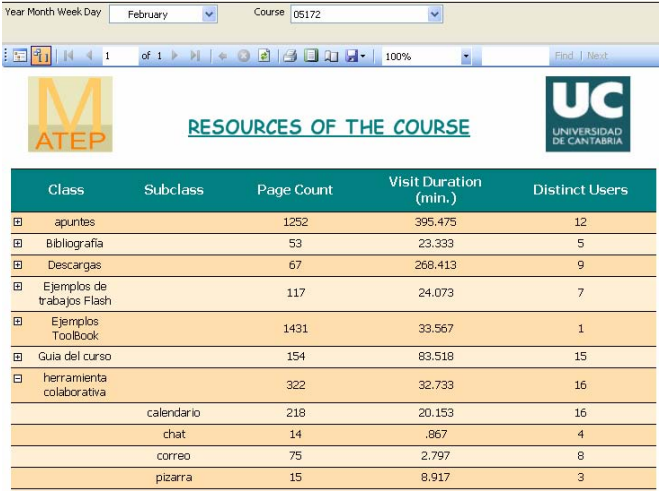


Fig. 4. Report about the usage of resources

In order to analyse the paths followed by the learners, the window in Figure 5 was developed. Teachers choose the initial page from which the analysis and the period of interest start and surf through the pages visited. For each one, some interesting information is shown such as the estimated time of study, the average and maximum time of stay, the number of times that the path has been followed and the number of students that did it.

This graphic acquires more sense if the course structure is designed by study sessions, as our case is. It means that each student must follow the steps indicated in the programmed sessions which combine theory and practice. The yellow lines represent pages which do not correspond to the chosen study session. So, the instructor can extract conclusions and redesign the course if he feels it to be necessary.

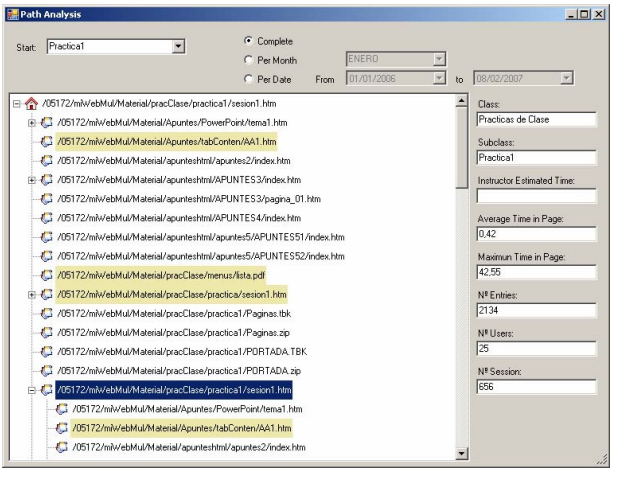
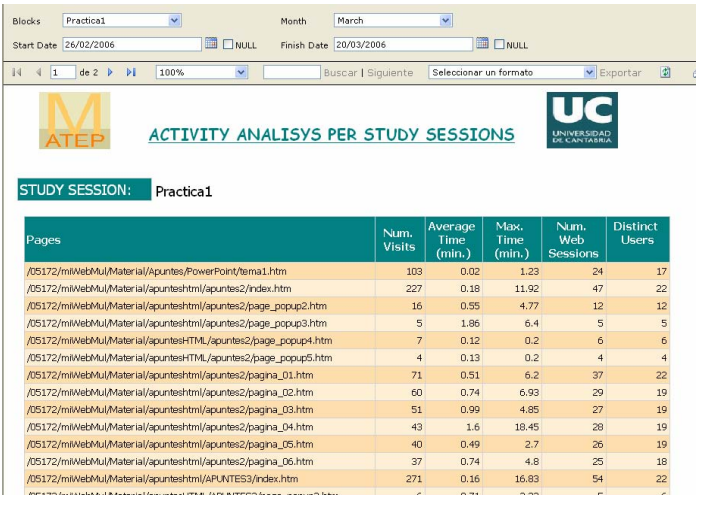


Fig. 5. Path analysis



Pages	Num. Visits	Average Time (min.)	Max. Time (min.)	Num. Web Sessions	Distinct Users
/05172/miWebMul/Material/Apuntes/PowerPoint/tema1.htm	103	0.02	1.23	24	17
/05172/miWebMul/Material/apunteshtml/apuntes2/index.htm	227	0.18	11.92	47	22
/05172/miWebMul/Material/apunteshtml/apuntes2/page_popup2.htm	16	0.55	4.77	12	12
/05172/miWebMul/Material/apunteshtml/apuntes2/page_popup3.htm	5	1.86	6.4	5	5
/05172/miWebMul/Material/apuntesHTML/apuntes2/page_popup4.htm	7	0.12	0.2	6	6
/05172/miWebMul/Material/apuntesHTML/apuntes2/page_popup5.htm	4	0.13	0.2	4	4
/05172/miWebMul/Material/apunteshtml/apuntes2/pagina_01.htm	71	0.51	6.2	37	22
/05172/miWebMul/Material/apunteshtml/apuntes2/pagina_02.htm	60	0.74	6.93	29	19
/05172/miWebMul/Material/apunteshtml/apuntes2/pagina_03.htm	51	0.99	4.85	27	19
/05172/miWebMul/Material/apunteshtml/apuntes2/pagina_04.htm	43	1.6	18.45	28	19
/05172/miWebMul/Material/apunteshtml/apuntes2/pagina_05.htm	40	0.49	2.7	26	19
/05172/miWebMul/Material/apunteshtml/apuntes2/pagina_06.htm	37	0.74	4.8	25	18
/05172/miWebMul/Material/apunteshtml/APUNTES3/index.htm	271	0.16	16.83	54	22

Fig. 6. Study session analysis

The report of Figure 6 offers more detailed information about the study session on a specific date. It allows the instructor to analyze when learners work and where they spend more time. This complements the report of Figure 7.

A very challenging task, for professors, is to find relationships between students and web-navigation behaviours. For that, a clustering of sessions must be carried out previously. In figure 7, the results obtained descriptively can be seen. We identify 4 clusters: the first gathers very short sessions in a regular day probably to read the news; the second it is the same as the first but staying more time (read /write mail, study contents, do activities, etc.); the third gathers longer sessions (48 min. on average) on days previous to the submission of a task; and finally, the fourth collects very

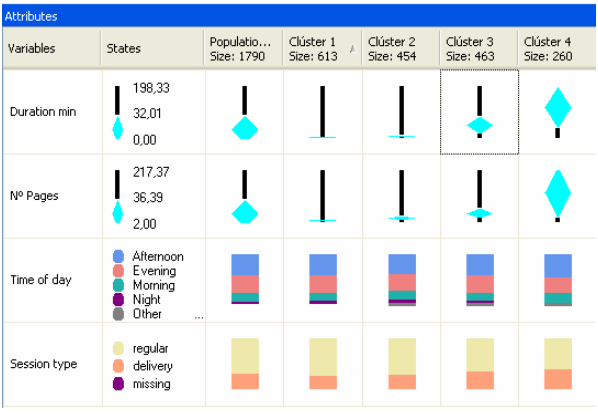


Fig. 7. Web-navigation behavior cluster

long sessions from students that work at the last moment (close to a submission deadline). It can be concluded that the learners' behaviour is similar to their behaviour in traditional teaching.

The product selected to develop the whole system is BI-SQL Server 2005.

4 Recommendations Based on Lessons Learned

Distance students require very explicit, well-organized and well-indexed web materials and resources. It is important to structure the entire course carefully before it starts and to be faithful to that structure throughout the course to provide students with a stable context in which to learn. We have checked that students navigate for the entire course in the first weeks and organize their work according to the submission dates. Likewise, these requirements are indispensable to do the course evaluation later. Besides, giving coherent names to files and organizing the course by study sessions is highly advisable. This will allow us to plot a more accurate study navigation path.

On the other hand, the context cannot be forgotten. Its importance is vital in order to understand the students' behavior. Events, news, exams, holidays, etc. influence their way of acting. So, all this information must be gathered and integrated. A good solution could be to store it in a data warehouse.

Finally, the success of the tool will be measured by its usability and the information that it provides. So it is very important to make a suitable selection of the indicators to be calculated and of the graphics with which these measures are shown.

5 Conclusions and Future Work

In this paper we present a tool, called MATEP, which provide dynamic reports and graphics components to track and assess the learning process in web-based platforms. It has been designed following the premises "easy to use" and "easy to interpret", with the aim that instructors obtain advantages of its use quickly.

Integrating MATEP with e-learning platforms will allow professors to analyze and visualize aggregated and detailed data, discovering student behaviour patterns', understanding how their courses are used, etc., that is, they will have in their hands quantitative and qualitative information with which they will be able to make better decisions.

As future work, we will analyze those attributes that best characterize the learning process and build significant patterns which can be used by a recommender agent. We will also endeavour to obtain an association map between students and navigation sessions.

References

1. Avouris, N., Komis, V., Fiotakis, G., Margaritis, M., Voyiatzaki, G.: Logging of fingertip actions is not enough for analysis of learning activities. In: Proc. Workshop Usage Analysis in learning systems (AIED 2005), Amsterdam (2005)
2. Gradío, A., Peris, R., Pinazo, D., Jiménez, A.: Features of student-lecturer interaction in e-learning. In: IV International Conference on Multimedia and Information and Communication Technologies in Education. Seville, Spain (2006)
3. Mazza, R., Dimitrova, V.: CourseVis: A graphical student monitoring tool for supporting instructors in web-based distance courses. International Journal of Human-Computer Studies 65(2), 125–139 (2007)
4. Merceron, A., Yacef, K.: Tada-ed for educational data mining. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning (2005)
5. Millán, S., Zorrilla, M., Menasalvas, E.: Intelligent e-learning platforms infrastructure. In: Proceedings of XXXI Latin American Informatics conference (CLEI), Cali, Colombia (2005)
6. Mostow, J., Beck, J., Cen, H., Cuneo, A., Gouvea, E., Heiner, C.: An Educational Data Mining Tool to Browe Tutor-Student Interactions: Time Will Tell!
7. Srivastava, J., Cooley, R., Deshpande, M., Tan, P.: Web usage mining: discovery and applications of usage patterns from Web data. SIGKDD Explor., vol. 1(2). ACM Press, NewYork (2000)
8. Tang, C., McCalla, G.: Smart recommendation for an evolving e-learning system. International Journal on E-Learning 4(1), 105–129 (2005)
9. Zaïane, O.: Building a Recommender Agent for e-Learning Systems. In: Proceedings of the International Conference on Computers in Education (ICCE) (2002)
10. Zorrilla, M.E., Menasalvas, E., Marín, D., Mora, E., Segovia, J.: Web usage mining project for improving web-based learning sites. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 205–210. Springer, Heidelberg (2005)
11. Zorrilla, M., Millán, S., Menasalvas, E.: Data webhouse to support web intelligence in elearning environments. In: Proc. of the IEEE International Conference on Granular Computing, Beijing, China (2005)

KRRT: Knowledge Representation and Reasoning Tutor System*

Jos A. Alonso[1], Gonzalo A. Aranda[2], and Francisco J. Martn–Mateos[1]

[1] Departamento de Ciencias de la Computacin e Inteligencia Artificial.
Escuela Tcnica Superior de Ingeniera Informtica, Universidad de Sevilla
Avda. Reina Mercedes, s/n. 41012 Sevilla, Spain
{jalonso,fmartin}@us.es,
[2] Dpto Ingeniera Electrnica, Sistemas Informticos y Automtica.
Escuela Politcnica Superior de La Rbida
Universidad de Huelva
Crta. Palos de la Frontera, Palos de la Frontera (Huelva), Spain
gonzalo.aranda@diesia.uhu.es

Abstract. Knowledge Representation & Reasoning (KR&R) is a fundamental topic in Artificial Intelligence. A basic KR language is First–Order Logic (FOL), the most representative logic–based representation language, which is part of almost any introductory AI course. In this work we present KRRT (Knowledge Representation & Reasoning Tutor). KRRT is a Web–based system which main goal is to help the student to learn FOL as a KR&R language.

1 Introduction

Knowledge Representation & Reasoning (KR&R) is a fundamental topic in Artificial Intelligence. A basic KR language is First–Order Logic (FOL), the most representative logic–based representation language, which is part of almost any introductory AI course. Usually students find difficulties in different aspects of using FOL as a knowledge representation and reasoning language. Two of them are about how to translate from natural language sentences into FOL formulae and how to construct formal proofs.

On the web "Logic software and logic education" [1], we can found several systems classified as logic educational software: Jape [4], "Logic Tutor" [2], Logic–ITA [6], "P–Logic Tutor" [7], OLIVER [10] among others. However, most of them are focused on how to construct formal proofs using natural deduction rules [4] or restricting themselves to propositional logic [2,6,7,10]. So, these systems are not concerned with how to use FOL as a KR&R language.

In spite of the amount of existing tools, there is an important subject on teaching logic for which these tools couldn't be used: formalization. Formalization is the process to translate arguments from natural language to logic. This is

* This research was funded in part by the Spanish Ministry of Education and Science under grant TIN2004–03884 and Feder funds.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 400–407, 2007.
© Springer-Verlag Berlin Heidelberg 2007

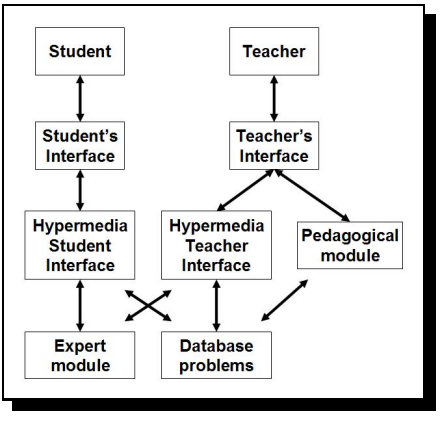
the initial step on "Computational Logic" subject and the main one when logic is used to represent knowledge. Our experience shows us that this is the most difficult process for students, and the way to learn it is solving exercises.

Until now, the only tools which help on teaching the formalization process are interactive forms as "Logic Tutor" by Huth and Ryan [5]. In these forms, the argumentations are presented and the student has to choose among the different proposals. As the forms have fixed answers, the possibilities of alternative and equivalent formalizations are excluded.

In this work we present KRRT (Knowledge Representation & Reasoning Tutor). KRRT is a Web–based system which main goal is help the student to learn FOL as a KR&R language. At knowledge representation stage, KRRT uses the FITS architecture's [3], based on the use of automated reasoning systems (OTTER [8], Vampire [9]) to prove the correctness of the formalization written by users. At reasoning stage, our tool makes a generalization of OLIVER [10] procedure, extending its reasoning power, from propositional logic to first order logic with equality.

2 Architecture and Implementation

KRRT uses the FITS architecture [3] showed in the following figure



The problem database is stored on MySQL. The teacher introduce new problems by an hypermedia application that generates the HTML forms displayed on the teacher interface.

The student interface is a set of HTML forms dynamically generated from the problem database by Apache web server.

The expert module is the section where the system intelligence is located. This module is the responsible to check all the teacher argumentation and the student answers.

2.1 Login into the System

KRRT is developed for being used on Internet. To enter it, the students must connect to the web page located at `http://www.cs.us.es/clg/krrt`. In the registration form the student is asked about her start level. Then, to analyze the student progress, information about the errors frequently repeated and the learning evolution is stored in the system. This is obtained maintaining the connection user and registering all the activities in the application on the time. This information could be used to estimate the hardness of the problem.

3 Problems Edition

The first KRRT service for teachers is to help on problem edition. To create a new problem, KRRT presents a page with 3 sections. The symbol section, the premise section and conclusion section; all of them with new, edit and delete button.

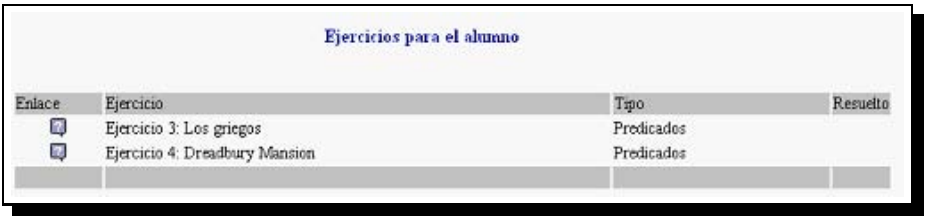
The teacher can create arguments and formalize them. The main window shows a list of all the arguments in natural language, together with the right formal equations written in first order logic language.



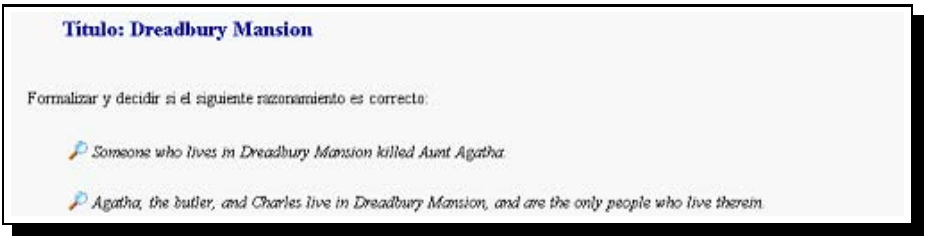
In the edition process, the expert module validates syntactically the formalization of each one of the sentences, and the correction of the whole argumentation. At the end, KRRT adds the new problem to its database to make it available for students.

4 Knowledge Representation Task

The first kind of task supported by KRRT is Representation. The system offers a collection of argumentation exercises (as it is shown in the following figure), and the student selects the one she wants to solve.

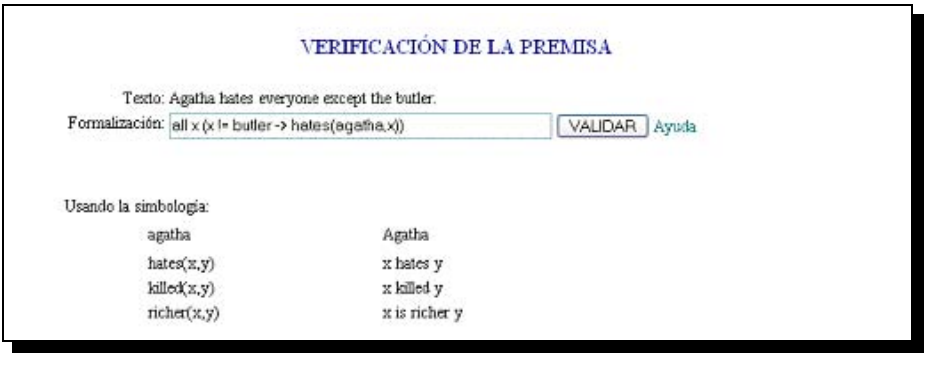


Once a problem is selected, a list of its hypothesis and conclusion is presented. If the student wants to solve the exercise, it must select, one by one, all of them and write the answers.

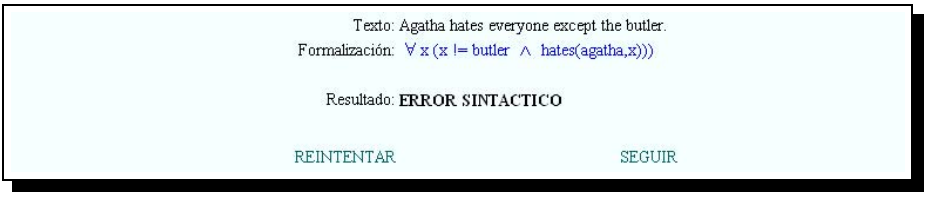




When an hypothesis is selected, the system shows the verification form. This form contains the hypothesis and a blank box where the student should write the formalization using OTTER syntax.



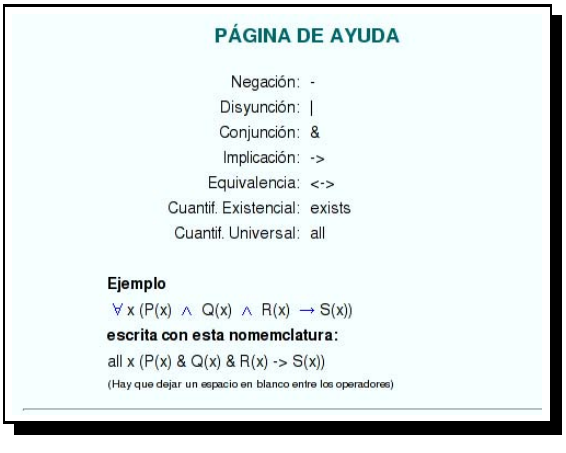
To check the formalization, the Validar button should be pressed and a new window appears with the result. The system reports syntax errors in the proposed formalization:



as well as successful formalization modulo semantic equivalence:

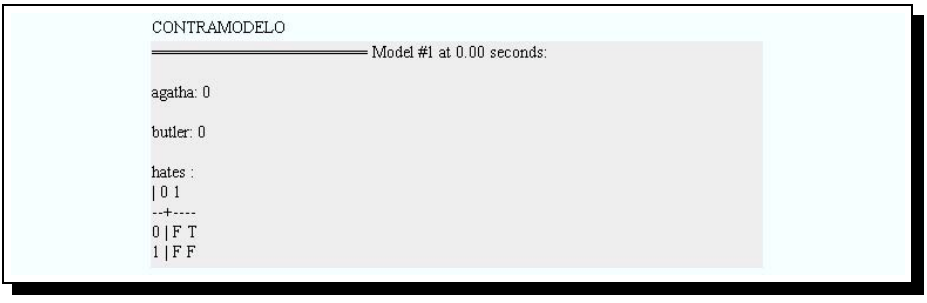


We note that the student formalization is showed in mathematical notation translating OTTER syntax. This happens whenever the system shows the equations.



The newest aspect of KRRT is its semantic nature; that is, the system accepts as right answer any formal equation logically equivalent to the solution, using the problem symbolization. In this example, the student could formalize the conclusion using the formula $\forall x\ [x = butler \vee hates(agatha, x)]$ or any other equivalent.

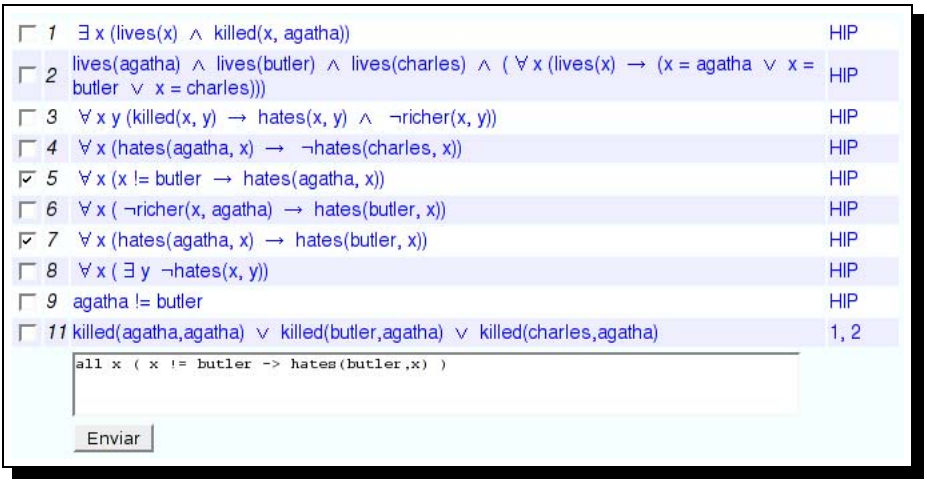
But if the conclusion is formalized as $\forall x\ [x \neq butler \rightarrow hates(agatha, x)]$ then the system indicates that it is an incorrect formalization and shows a counter-model.



5 Reasoning Tasks

The second kind of task supported by KRRT is Reasoning. When the student finishes to correctly formalize an exercise, then it is added to the reasoning exercises list available for her.

For example, suppose that the student has correctly formalized the Agatha exercise and she wants to start to reasoning on it. In this way, she chooses it from the exercises list and the system shows a form page with the whole formalization and a text box. Here, the student have to click a maximum of two choices and write the formula which she thinks is a logic consequence from the clicked ones.



When the formula is proved, KRRT add it to the premise list, indicating what formulae had been used to deduce it. Also this new formula can be used to deduce new formulae; in the other hand, if the formula cannot be proved, the system shows an counter-example.

To finish the exercise, the student must press the Finalizar button, then KRRT checks semantically if the last formula written is equivalent to the exercise conclusion. If it does, a message is showed notifying that the answer is right. Otherwise, a counter-example is presented.

6 Other Services of KRRT

KRRT offers other services for teachers, as the student history and its evaluation. Also it can do an evaluation about the hardness of the problems. These services are being incorporated currently.

To students, the system offers services like exercise status, which it can be one of these: Not touched, started with errors (syntax or semantics ones) and corrects (finished or not)

Also a pedagogical module is being connected, to extract conclusions that help us to identify the student difficulties in a more detailed way.

7 Conclusions and Future Works

In this work, we present KRRT, the first intelligent tutor for teaching formalization with semantics correction.

Our future works are leaded in two lines: to incorporate KRRT during the present course 2006–07 on teaching the subject of "Computational logic" and to enlarge their services, from the point of view of their automatization, adaptation and evaluation.

References

1. Logic software and logic education, `http://www.cs.otago.ac.nz/staffpriv/hans/logiccourseware.html`
2. Abraham, D., Crawford, L., Lesta, L., Merceron, A., Yacef, K.: The Logic Tutor: A multimedia presentation. Electronic Journal of Computer–Enhanced Learning (October 2001)
3. Alonso, J.A., Aranda, G.A., Martín.-Mateos, F.J.: FITS: Formalization with an Intelligent Tutor System. In: IV International Conference On Multimedia And Information And Communication Technologies In Education (2006)
4. Bornat, R., Sufrin, B.: Jape: A calculator for animating proof-on-paper. In: McCune, W. (ed.) Automated Deduction - CADE-14. LNCS, vol. 1249, pp. 412–415. Springer, Heidelberg (1997)
5. Huth, M., Ryan, M.: LICS web tutor, `http://www.cs.bham.ac.uk/research/projects/lics/tutor`
6. Lesta, L., Yacef, K.: An intelligent teaching assistant system for Logic. In: Cerri, S.A., Gouardéres, G., Paraguaçu, F. (eds.) ITS 2002. LNCS, vol. 2363, pp. 421–431. Springer, Heidelberg (2002)
7. Lukins, S., Levicki, A., Burg, J.: A tutorial program for propositional logic with human/computer interactive learning. In: SIGCSE 2002, pp. 381–385. ACM, New York (2002)

8. McCune, W.: OTTER 3.3 reference manual. Argonne National Laboratory (2003)
9. Riazanov, A., Voronkov, A.: Vampire 1.1 (system description). In: Goré, R.P., Leitsch, A., Nipkow, T. (eds.) IJCAR 2001. LNCS (LNAI), vol. 2083, pp. 376–380. Springer, Heidelberg (2001)
10. Wildenberg, A., Scharff, C.: OLIVER: an OnLine Inference and VERification system. In: 32 ASEE/IEEE Frontiers in Education Conference, IEEE (2002)

Fuzzy Adaptive Objects (Logic of Monitors as Agents)

Germano Resconi[1] and Javier Alonso[2]

[1] Catholic University, via Trieste 17,
Brescia, Italy
resconi@numerica.it
[2] Object Oriented Technology Laboratory, Department of Computer Science,
University of Oviedo, Calvo Sotero s/n
33005 Oviedo, Spain
javier@a4devis.com

Abstract. The active semantic in Adaptive Object-Model (AOM) is only one example of the more complex active semantics at different orders. When we violate the integrity of the system, uncertainty grows up in the system. When monitors are conflicting agents in the modal logic, we can study uncertainty with the logic of the monitors that is comparable with the logic in the fuzzy set theory. Fuzzy values (integration degree) of a concatenation of interactive objects can be computed by fuzzy AND, OR and NOT operators presented in this paper. A short presentation of RDM framework suggests how it is possible to use object oriented language to implement adaptive object modelling. At the same time, relationship extension at different orders and also violation of the integrity of the system (uncertainty), where we break the constrain conditions; can be studied with RDM.

1 Introduction

An Adaptive Object-Model (AOM) is a system that represents classes, attributes and relationships as metadata. Users change the metadata (object model) to reflect changes in the model. These changes modify the system's behaviour.

The relation among objects or interactive object generates constrains that we control by the monitors and propagators. To enforce constrains requires that the related objects were updated with information describing the trigger when the propagator is instantiated. Active semantics uses monitors for a local object. For a far object to actively restore the integrity of the interactive object we use the propagator that propagates through objects the message of the monitors. The active semantics in AOM is only one example of the more complex active semantics at different orders. When we violate the integrity, the monitors give the information how and where the integrity is violated. When we assume that any monitor is a agent in the modal logic, the monitors are a set of conflicting agents. With the logic of the monitors we can study how uncertainty can be computed by the AND, OR and NOT elementary logic operators. The logic of the monitors can be compared with the logic of the fuzzy sets. We create logic expressions with the monitors and we compute the degree of integrity in complex logic situations.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 408–414, 2007.
© Springer-Verlag Berlin Heidelberg 2007

An example of implementation of active semantics with object oriented language is the RDM framework.

2 Entity and Relationship

The Entity –Relationship structure follows the Specialization Principle in the RDM Framework. Any entity has two roles that must be coherent with the entity itself. Thus, the Entity – Relation can be modelled in this way

$$M = < ROLE , RELATION , ENTITY , F , G > \quad (1)$$

Where ROLE is the set of roles, RELATION is the set of relations, ENTITY is the set of entities and

$$F : ROLE \times RELATION \rightarrow ROLE \quad (2)$$

Is the transition rule for which given the "relation" we can obtain a role from another role. The function G

$$G : ROLE \times RELATION \rightarrow ENTITY \quad (3)$$

Is the reply rule for which we can associate any role and relation with an entity.

Because any entity is not a simple element but is an abstract element with different instances, we can describe the internal structure of the entity by the rule

$$H : INSTANCE \times ATTRIBUTE \rightarrow VALUE \quad (4)$$

Where, instances are samples of the Entity class.

3 Active Semantics

Relationships capture the semantics of the interactive objects and become the means by which active semantics is used to express behavioural composition.

Adaptive Object Modelling (AOM) is a model based on instances rather than classes. When we define the object type class, any instance of the class is an instance of the object. With the introduction of the object type, we can have active objects and passive objects. In passive objects we have only the reaction of the object to a message.

On the other hand, every reactive object can be represented in this synthetic way.

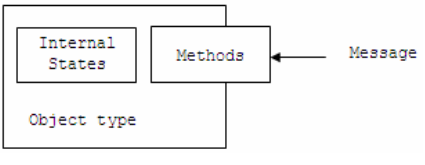


Fig. 1. Internal structure of one object in the object oriented language

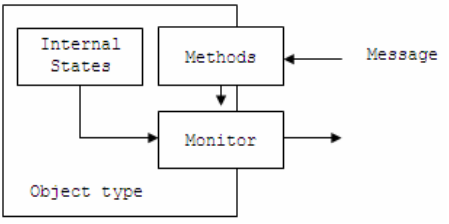


Fig. 2. Internal structure of an active object with monitor

Adding relationships (Participants) to an Active Object can be shown as follows:

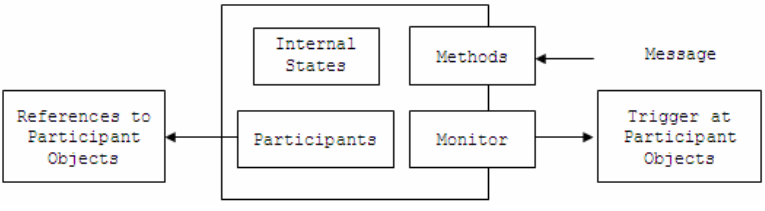


Fig. 3. Internal structure of the interactive object with monitor and participants in the relationship Feedback by active semantic

So far, the way to guarantee the integrity of the system in RDM follows the Externalization Principle, every change in the model's state must produce a notification (triggering). This is enhanced by applying RuleObjects to Monitors and Reactions in order to model rules with more complex algorithms. There may be a rule that asserts that certain entity-relationships are only legal if the entities have certain values and other constraints are met.

4 Monitors as Logic Agents in the Meta-theory of Uncertainty

We know that the agent (monitor) is the entity by which we can know if a proposition or assertion is TRUE of FALSE. When the enforcement rule (assertion) is violated the assertion is FALSE and the integrity of interactive objects is not valid.

In the Adaptive Object Modelling the implied action of an assertion is to reject any action which would lead to violation of the constraint. But in general we assume that the implied action cannot always eliminate the violation of the constraint. In this case we break the integrity and the system has uncertainty condition. When the assertions are given as TRUE in all the monitors the associated objects are completely integrate and the constrain condition is valid. When all the monitors give the value FALSE then we have a complete violation of the integrity. But, when for a part of the monitors the assertions are true and for other parts the assertions are false we have a partial integrity of the objects. In conclusion we assume that:

– We associate to a monitor a logic agent
– The relation among the monitors is a relation among logic agents

– We associate to any assertion in a monitor (logic agent) a logic value TRUE or FALSE

With the monitors (agents) we define the *model*

$$M = < W, R, V > \tag{5}$$

where W is a non-empty set of logic agents (monitors), R is any type of relation among the agents. $R \subseteq W \times W$ is an accessibility relation on W, and V is the function that assigns a logic value TRUE or FALSE to any monitor.

$$V: \text{Propositions} \times W \rightarrow \{T, F\} \tag{6}$$

Example : The function V for five agents can be represented in this way

$$V(agents) = \begin{pmatrix} Agent_1 & Agent_2 & Agent_3 & Agent_4 & Agent_5 \\ False & True & True & False & False \end{pmatrix}$$

A proposition p is necessarily true when in all the accessible agents (monitors) p is true. The proposition p is necessarily false when there is at least one accessible agent (monitor) where p is false. The proposition p is possibly true when there is at least one accessible agent (monitor) where p is true.

Resconi, *et al.* (1992-1996) suggested to adjoin a function

$$\Psi : W \rightarrow R \tag{7}$$

where R is the set of real numbers assigned to agents W in order to obtain the new model

$$S1 = < W, R, V, \Psi > \tag{8}$$

That is for every agent, there is an associated real number that is assigned to it. With the model S1, we can build the *hierarchical meta-theory* where we can calculate the expression for the membership function in the fuzzy set theory .

Imprecision means that an “entity” (temperature , velocity …) cannot have a crisp logic evaluation. monitors, i.e. agents. In this case a agent is associated to a monitor in the active object.

The violation of the integrity in an interactive object is the principal source of the imprecision in the meaning representation of the interactive object.

When we write for short:

$$\mu_{p_A}(x) = \frac{(set\ of\ \text{monitors where } p_A(x) \text{ is true})}{|W(x)|} \tag{9}$$

where pA(x) is the integrity attribute for the interactive object x in the set A of interactive objects, the variable $\mu_A(x)$ is the membership function in the fuzzy set A of the interactive objects and for a particular interactive object x.

The membership expression is computed as the value of Ψ in S1 stated in (1) above. It is computed by the expression

$$\Psi = \frac{1}{|W|} \tag{10}$$

5 Logic of the Monitors and Fuzzy Set Theory

Because any monitor as a agents gives us information where we violate the integrity of the system, we are interested in the development of logic operations by which we can create logic expressions in the logic of the monitors.

5.1 Operation AND

Starting from two propositions, such as ' "John is tall" is true' AND "'John is heavy" is true', given that we know the membership functions of $\mu_{p_1}(x)$ for P_1: "'John is tall" is true', and $\mu_{p_2}(x)$ for P_2: "'John is heavy" is true'. It should be clear that if we know the set of possible agents $W_1=\{Wi\}$ where p_1 is true and the set of possible agents $W_2=\{Wj\}$ where p_2 is true, then we can compute

$$\mu_{p_1}(x) = |W_1| / |W| \text{ and } \mu_{p_2}(x) = |W_2| / |W|. \tag{11}$$

Because $|W_i| = N_i$ is the number of possible agents where the proposition p_i is true. We generate a new event p such that "p_1 and p_2" is "true" by the expression

$$p = p_1 \wedge p_2 \tag{12}$$

where "and" is interpreted to be equivalent to "$\wedge$" operation. So we have

$$\mu_p(x) = \mu_{p_1 \wedge p_2}(x) = \frac{|W_1 \cap W_2|}{|W|} \tag{13}$$

Remark 1. When we know the value of p_1 and p_2 for every agent we evaluate the expression $p = p_1 \wedge p_2$. Because for p_2 I can choose any type of sentence, we can choose $p_2 = \neg p_1$. When the set of agents where p_1 is true and the set of agents where p_2 is true have intersection different from zero, irrational agents (monitors) can grow up: obtaining agents where $p = p_1 \wedge \neg p_1$ is true.

We can easily show that for W_1^C, the complement of W_1 we have

$$\mu_{p_1 \wedge p_2}(x) = \frac{|W_1 \cap W_2|}{|W|} = \frac{|W_2|}{|W|} - \frac{|W_1^C \cap W_2|}{|W|} = \min[\mu_{p_1}(x), \mu_{p_2}(x)] - \frac{|W_1^C \cap W_2|}{|W|} \tag{14}$$

In this case we have W1∩W2 = W2 that is the set with the minimum value of cardinality.

When p2 = ¬ p1 we have that all the agents in W2 are irrational. We can prove that

$$0 \le \mu_{p_1 \wedge p_2} \le \min(\mu_{p_1}, \mu_{p_2}) \tag{15}$$

Between a zero irrationality to the maximum of the irrationality in the monitors.

5.2 Operation OR

For the OR combination, following similar steps as in the paragraph above, we have:

$$\max(\mu_{p_1}, \mu_{\neg p_1}) \leq \mu_{p_1 \vee \neg p_1} \leq 1 \tag{16}$$

In this case, the set where $\neg$ p1 is true is included in the set where p1 is true, we break the classical property for which the set of agents where $\neg$ p1 is true is the complement set of the agents where p1 is true.

5.3 Operation NOT

In the fuzzy calculus we break the classical symmetry for which:

$$\mu_{\neg p} = 1 - \mu_p \tag{17}$$

The set where p is true and the set where $\neg$p is true are separate sets without any connection one with the other as we have in the classical modal logic. Now with the irrational variable IR , we can extend the negation operation in this way

$$\mu (\neg X) = \mu (\neg X \oplus IR) \qquad (30)$$

Where the operation "$\oplus$ " is the XOR. The proposition IR is the irrational sentence that is true when $\neg X \wedge X$ is True or when $\neg X \vee X$ is False. Because in the crisp set and in the classical logic we have always that $\neg X \wedge X$ is False and $\neg X \vee X$ is True, we have that always IR is False. In this case we have the expression (29).

6 Conclusion

This paper extends the traditional relationship considered as first order relationship to a more complex realtionships at higher orders. The new type of realtionship includes the monitors as instruments to establish the integrity of the different relationships at the second, third or higher orders. The network of the new type of relationships generates objects which have methods that are not equal but similar. We remark that the monitors trigger actions by which dynamically the similarity is obtained as a constraint or integrity. In the traditional entity/relation a constraint is created. For more complex realtionships more complex constraints are introduced to establish internal coherence of the realtionship itself. Partial coherence or integrity is introduced. When we have defect in knowledge, we violate the integrity of the system of interacting objects. The monitors give us the position in the system where the integrity is violated. Meta-theory of uncertainty by modal logic, where the agent is a monitor, can generate a special logic or logic of the monitors by which we can compose the monitors results with the AND, OR and NOT logic operations.

References

1. Rescon, G., Murai, T.: Field Theory and Modal Logic by Semantic Field to Make Uncertainty Emerge from Information. Int.J.General System (2000)
2. Resconi, G., Türkşen, I.B.: Canonical forms of fuzzy truthfulnesss by meta-theory based upon modal logic. Information Sciences 131, 157–194 (2001)
3. Resconi, G., Klir, G.J., St, U.: Hierarchical Uncertainty Metatheory Based Upon Modal Logic. Int. J. of General Systems 21, 23–50 (1992)
4. Izquierdo Castanedo, R.: RDM: Arquitectura software para el modelado de dominios en sistemas informáticos. Doctoral Thesis (2002), http://di002.edv.uniovi.es/ ric/tesis.pdf
5. Germano, R., Jain Lakhmi, C.: Intelligent Agents (Theory and Applications). Springer, Heidelberg (2004)
6. Joseph, W., Yoder, R.J.: The Adaptive Object-Model Architectural Style, WICSA3 2002, http://www.adaptiveobjectmodel.com/WICSA3/ArchitectureOfAOMsWICSA3.pdf

E-Learning Platform as a Teaching Support in Psychology

Alberto Salguero[1], Francisco Araque[1], Antonio Fernández-Parra[2], M. Isabel Jiménez[2], and M. Carmen Vives[2]

[1] Department of Software Engineering, University of Granada (Andalucía), Spain
[2] Faculty of Psychology, University of Granada (Andalucía), Spain
{faraque,agsh,afparra,mjimenez,mvives}@ugr.es
Universidad de Granada, 18071, Granada (Andalucía), Spain

Abstract. Increasingly, e-learning platforms are being implemented at various educational levels as a form of teaching support. In addition to promoting flexibility in teaching and to permitting the incorporation of new educational materials and resources, e-learning aims to familiarize students with the use of the computer as a work tool. The present paper describes an experiment involving the construction and adaptation of an e-learning platform to the teaching of psychology at university level. The paper presents findings concerning the use of the platform by students and the relation between this use and academic performance in two courses over two complete academic years. Results show that students who take advantage of the virtual educational platforms generally present a higher level of participation and academic achievement.

Keywords: teaching/learning strategies; postsecondary education; improving classroom teaching; intelligent tutoring system.

1 Introduction

In recent years, university education has been undergoing many changes, most notably the increasing demand for higher quality in teaching and for greater flexibility in structure and teaching possibilities. Additionally, competition between university centres is increasing and territorial diversification is taking place on a larger scale. For several years there has been an increase in the number of students who are unable to follow the traditional format of university education, owing to the time and space constraints inherent in regular attendance at traditional classes. At the same time, teaching methods are also undergoing a transformation, with a greater emphasis on student-centred learning, enabling the student to acquire more direct control over his or her learning rather than depending exclusively on the lecturer.

In view of recent technological developments, new societal demands and the changes in higher education mentioned above, there is a need to design and apply new computer programs aimed at enhancing the flexibility of teaching with regard to time and place, and at adapting courses to the individual needs of each student [1]. Computer technology is increasingly viewed as an important teaching tool and as a means to enhancing education [2]. E-learning technology has evolved to provide users with

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 415–422, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the specific tool and information needed at the point of activity, and allows a move away from a fully pre-scheduled curriculum to provide demand and open access. The internet has developed to an extent where it is possible to use the web for didactic purposes, opening up new possibilities for the development of different teaching strategies [3]. As a consequence of the introduction of new technologies in the field of education, the scenario of educational institutions is also changing. Virtual webs are breaking down the traditional isolation of the classroom, eliminating barriers of space, time, identity and status, and bringing about a greater emphasis on student-centred learning [1].

In recent years, there has been an increase in the number of e-learning program applications in the field of education. In [4] is presented a web system which allow self-assessment by the student and offer immediate feedback concerning his or her results. In [5] a students' experience of an e-learning program for a Master's Degree in Taxation (Master Program in Taxation) is described. This program also offered feedback to students concerning their progress and provided personalized assessment over brief periods of time. Other universities and educational centres have developed e-learning platforms as a central part of their teaching resources, including Maryland University, Western Governors University, the Oberta University in Catalonia and the CISCO corporation. Similarly diverse are the areas of knowledge to which e-learning platforms have been applied, from A Course in Microbiology [1] to Teaching Mathematics and Statistics [6].

In psychology, application of educational resources via e-learning platforms is especially interesting since evaluation and treatment resources are increasingly computer-based. However, while the number of published programs is rising exponentially year by year, present use of information technology by psychologists remains at a level similar to that of 1987 [7] [8]. In the view of some authors, this phenomenon is due to the lack of adequate training and instruction in information technology for practising psychologists [8]. With this in mind, the motivation for the present experiment has been to incorporate an e-learning platform as part of the practical teaching of psychology in order to achieve the following goals: (i) to substitute physical attendance in some tasks and/or time periods for students who cannot attend classes regularly for reasons of health or work; (ii) to complement face-to-face teaching with internet resources incorporated in the platform; and (iii) to introduce students to the use of information technology in psychology.

In the present paper we shall describe the platform designed for this purpose and present findings relating to its application over two consecutive academic years in two courses which form part of the BA Honours Degree in Psychology at the University of Granada (Andalucía), Spain.

This paper is organized as follows: next section introduces the e-learning platform. In third section a case study and some experimental results are presented. In forth section discussions about the experiment are presented. Finally, we end with conclusions and future work.

2 E-Learning Platform Description

In this section, we shall present the main characteristics and services offered by e-learning system. In our case we have chosen to use ASP.NET which is part of the

Microsoft.NET framework [9]. It allows the creation and the use of distributed Web Services and supplies of accessible business from multiple platforms through the Internet without taking into account of its technical implementation.

Like many other e-learning applications, ours is based on learning modules. Each module contains a set of didactic materials and an evaluating method. Instructors can offer and learners can sign up for whole courses that are scheduled to run over a fixed duration of time. Courses are structured by combining a number of didactic modules. These modules can be selected to be visible for students from a repository at any time. Students can select the modules that wish to work on. In order to develop these modules the collaboration of several students may be necessary. The result of these must be sent to the professor through this tool. This mechanism saves the professor the work of collecting the students assignments, by creating an automatic control of dates of delivery.

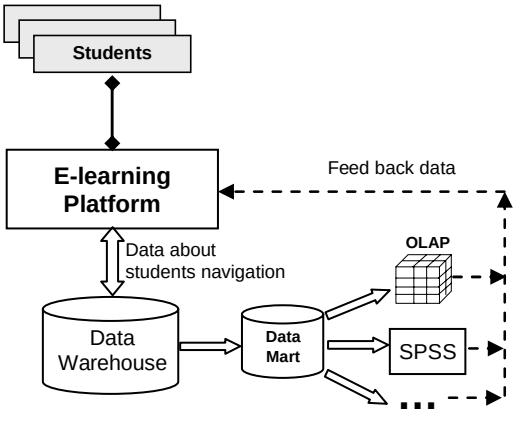


Fig. 1. Functional Architecture

Our application allows the use of MCQ (multiple-choice questions) as the basic evaluating method of a module, together with a more traditional one that allows the student the submission of complete projects or case studies. After evaluating these projects or case studies the teachers can make observations to each student individually.

There are two kinds of statistics that the system can manage: access statistics and participation statistics. The access statistics section shows the flow between different pages of the application and the clicks that have been done in the same page. We can obtain access statistics by month, week, day, hour and sections, for every user. The participation statistics section allows the teacher to know how the students are using the system. Each click of the students that generates an abstract action is, thus, recorded (post a message in the forum, send a file, etc). We can generate participation statistics by month, week, day and hour, for every action performed by a specific user. The system is able to make a comparison between a single student and the whole group.

We can see the functional architecture in figure 1. As we can observe the e-learning platform uses a Data Warehouse to store all information about user, lessons, modules, mouse clicks, etc. It is possible to generate personalized reports by means of OLAP (On-Line Analytical Processing) functionalities as well as use the data to feed other statistical analysis software like SPSS. The OLAP functionalities are a set of built-in modules which facilitates the use of typical OLAP operations over the data inside the e-learning platform by the teacher at any moment.

3 Description of Experiment

The objective is to ascertain the use of the e-learning platform made by students of psychology and to establish the relation between such use and students' academic commitment and performance in the courses concerned.

A total of 597 students of the Faculty of Psychology at the University of Granada participated in the study. Of these, 72.2% were students following a compulsory third year course (Psychological Evaluation) and 27.8% were students following an optional fourth-year course (Clinical Child Psychology). 83.7% of students were female and 16.3% were male. Only 0.7% had jobs in addition to their studies. 3.1% were over thirty years of age.

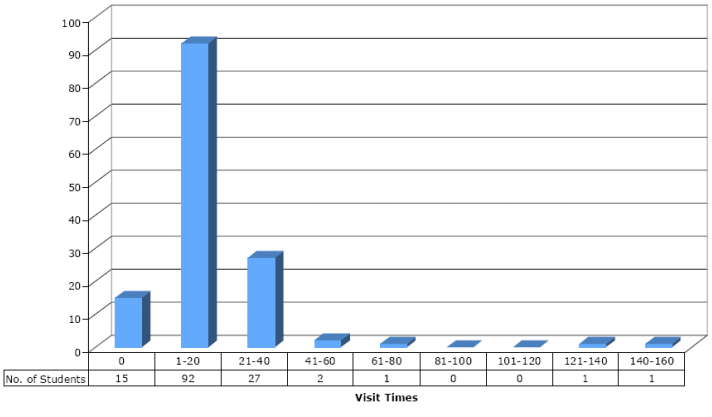


Fig. 2. Visit Times

The opportunity to carry out course activities on the internet via the e-learning platform was offered to students of Psychological Evaluation in 2003-04 and to students of Clinical Child psychology in 2004-05. Activities included consultation of the program, texts and results as well as carrying out some of the practical course assignments. At the end of each academic year analysis was undertaken of students' results, participation (in terms of number of visits to the web page) and use of the platform to carry out and hand in the practical tasks. Statistical analysis consisted of an ANOVA one-factor analysis (academic performance) for visits, Chi-square and Student's T-test for independent inter-group samples (virtual/class-based) and separate variables for commitment, performance and gender. The SPSS 12.0 program was used in all analyses.

Concerning to the use of the platform, of the 597 students enrolled in the courses, a total of 139 (23.28%) registered as users of the platform and 103 (17.25%) used the platform to carry out the practical tasks. In all, the platform was used 3,214 times by students. In spite of having access to the platform, 15 students (10.79%) never made use of it, while the remaining students visited the platform at least once (see Figure 5 for details). Most students used the platform between 1 and 40 times (85.61%); however, one student used the platform 134 times and another as many as 146 times. A simple consultation of the main page of the platform (notice board) did not count as a visit, since this permitted universal access with no registration requirement.

Some students reported that they did not use the platform owing to problems such as not owning a computer, forgetting their password, lack of training in information technology and lack of confidence in the tool as a means of communication with the lecturer.

In general, no relation was found between the number of visits (Figure 2) to the platform by students who used it to carry out the practical tasks and academic results. The group of students who failed the courses presented a mean of 12.46 visits to the site (Standard Deviation = 12.4), while the successful group presented a mean of 19.28 visits (Standard Deviation = 21.38). Anova factor analysis of these data did not show significant differences between the two groups (F1,58 = 1.199, N.S.).

Table 1. Comparison of results in the various course assessment tests between students who carried out practical tasks on the e-learning platform and classroom-based students

	Group*			
Variable	Virtual	Class-based	*t*	*p*
First partial exam results (out of 7)	4.38 (1.24)	3.72 (1.39)	3.15	0.002
Second partial exam results (out of 7)	4.17 (1.31)	3.22 (1.50)	4.12	0.0001
Results for practical tasks (out of 3)	1.94 (0.76)	1.50 (0.64)	4.93	0.0001
Final grade (out of 4)	1.84 (1.51)	1.19 (1.20)	3.98	0.0001
Practical tasks undertaken (out of 6)	4.66 (1.41)	4.32 (1.43)	1.76	Ns

* Mean (standard deviation).

Comparison between virtual and classroom-based students was made of results obtained by virtual group students who used the system at all levels, including carrying out the practical tasks, and students from the classroom-based group. For this an independent sample Student T test was applied. As shown in Table 1, the groups are different with respect to all variables concerning performance in the exams and practical tasks. The virtual group students obtained better results in all assessments undertaken throughout the two courses. Results for these students were significantly superior in both the first theoretical exam (also known as the first partial exam) and in the second theoretical or partial exam, as well as in assessment of the practical tasks and in the final grades for the two courses. Although no significant inter-group differences were revealed with regard to the number of practical tasks undertaken, students who carried out the practical tasks on the e-learning platform achieved better results than classroom-based students in all assessment tests, including the practical tasks themselves.

The virtual and classroom-based groups were also compared by means of the Chis-quare test for variables such as gender, course followed, completion or failure to

complete all assessment tests and final grade obtained (pass or fail). Results for these variables are shown in Table 2. As may be seen, no significant differences were found between the two groups with regard to the distribution of male and female subjects. In both groups, the number of women was far higher than that of men; specifically, 78.64% of the virtual group and 84.30 % of the classroom-based group were women. With regard to course followed, in Psychological Evaluation students using the e-mail platform represented 17.4 % of the total, while the proportion of students using the platform in Clinical child psychology, taught the following academic year, was 16.86% These differences were not significant according to the statistical analyses undertaken.

Table 2. Results of chi-square comparisons between virtual and classroom-based groups with regard to gender, course, completion of assessments and final grades

	Group*			
Variable	Virtual	Class-based	x^2	*p*
Gender				
Female	81	419		
Male	22	75	2.389	Ns
Course				
Psychological Evaluation	75	356		
Clinical Child Psychology	28	138	0.023	Ns
Completion of assessments				
Yes	47	269		
No	56	255	2.662	Ns
Final Grade				
Fail	11	130		
Pass	36	139	10.645	0.001

* Number.

No differences did arise with regard to the number of students who undertook all the assessments (theoretical exams and practical tasks) in the two groups. Specifically, 45.63% of the virtual group completed these assessments, while the percentage for the classroom-based group was 54.45%. However, significant inter-group differences were found concerning the final result for the courses. In the virtual group, 76.59% of students that completed the assessments passed the course, while the percentage for the classroom-based group was considerable lower, at 48.32%.

4 Discussion

The present study was carried out to assess the incorporation of an e-learning platform as a part of the teaching of psychology in a Spanish University. All the students could have an account in the system and could access to the services of the e-learning platform. Even though, only 23.28 % of the students registered as users of the platform, and only 17.25% used it for the realization of the practical tasks. Apparently, the lack of ability for the use of the information technology resources could be one of the factors that influenced the election. On the other hand, most of the students were able

to do the practical tasks in the classrooms (traditional way) so the did not need to use de e-learning platform. For this reason, it can be concluded that the level of use of the system is acceptable, bearing in mind that one of the goals of this experiment was to replace physical attendance to practical tasks for the students who were not able to attend classes regularly.

As we have seen, the average number of visits to the e-learning platform by psychology students who had gained access to the system was 15. Given that the platform was in continual operation during the nine months of each academic year, this figure reflects a low rate of use. This finding would seem to confirm results obtained by other authors who point out that use of these resources by university students is still limited and that on many occasions students continue to have misgivings about the use of information technology and to experience difficulty in adapting to new and more active and independent learning environments [10].

The option of this 'semi-virtual' system of teaching was taken up similarly by male and female students. No significant differences were detected between the two courses, although the fact that there was a slight increase in the percentage of users in the second course may be interpreted as a consequence of students' greater familiarization with the tool. This would be coherent with the finding described in various studies, that students' previous experience is a predicting factor for success when using this type of resource. In this sense, some studies have concluded that one factor which predict that students will achieve a good level of virtual interaction is that they have experience and training with new technologies [11].

It also seems significant that while both virtual and classroom-based groups completed a similar number of assessment tests and practical tasks – a predictable result, given that the practices were a compulsory part of both courses –, the students who used the platform obtained better results.

This result may be interpreted as showing that the students who opt for the virtual tool have higher motivation, commitment and/or interest towards the learning process, and therefore make use of all the learning resources available to them. Bearing in mind that use of the platform was completely voluntary, this interpretation would seem to explain the superior final results obtained by the virtual group. Moreover, such a view would be in line with [12], who argue that much of what happens when e-learning platforms are used depends on the students' own motivation.

5 Conclusion

To sum up, we may conclude that at present, psychology students make scant use of the e-learning resources available to them, although an effect of familiarization does seem to occur if such tools are incorporated as part of the teaching programme. Moreover, students who do take advantage of these systems obtain significantly better results. Nevertheless, for next studies it could be taken into consideration the training of the students in new virtual technologies before the implementation of the e-learning platform; which may improve the results found in the present investigation.

This work has been supported by the Spanish Research Program under project TIN2005-09098-C05-03 and by Andalucía Research Program under project 2006/282916.

References

1. Masiello, I., Ramberg, R., Lonka, K.: Attitudes to the application of a webbased learning system in a microbiology course. Computers & Education 45, 171–185 (2005)
2. Chyun, S.Y., Vachon, M.: An investigation of the Profiles of Satisfying and Dissatisfying Factor in E-Learning. Performance Improvement Quarterly 18, 97–113 (2005)
3. Evans, C., Gibbons, N.J., Shah, K., Griffin, D.K.: Virtual learning in the biological sciences: pitfalls of simply putting notes on the web. Computers & Education 43(1-2), 49–61 (2004)
4. Sung, Y.T., Chang, K.E., Chiou, S.K., Hou, H.T.: The design and application of a Web-based self- and peer-assessment system. Computer & Education 45, 187–202 (2005)
5. Engelbrecht, E.: Adapting to changing expectations: Post-graduate students experience of an e-learning tax program. Computer & Education 45, 217–229 (2005)
6. Stefansson, G.: The tutor-web: an educational system for classroom presentation, evaluation and self-study. Computers & Education 43, 315–343 (2004)
7. McMinn, M.R., Buchanam, T., Ellens, B.M., Ryan, M.K.: Technology, professional practice and ethics: survey findings and implications. Professional Psychology: Research and Practice 30, 165–192 (1999)
8. Olson, K.R.: Computerized psychological test usage in APA-accredited Training Programs. Journal of Clinical Psychology 57(6), 727–736 (2001)
9. Tapadiya, P.K.:NET programming: a practical guide using C#. Prentice Hall PTR, Upper Saddle River, N.J. (2002)
10. Dewhurst, D.G., Macleod, H.A., Norris, T.A.: ndependent student learning aided by computers: an acceptable alternative to lectures? Computer & Education 35(3), 223–241 (2000)
11. Motiwalla, L., Tello, S.: Distance learning on the internet: an exploratory study. Internet and Higher Education 2, 253–264 (2000)
12. Lin, B., Hsieh, C.: Web-based teaching and learner control: a research review. Computers & Education 37, 377–386 (2001)

Wiki Use in Learning for Topography Spanish Students

Ignacio González Alonso, Mercedes R. Fernández Alcalá, Benigno Villa Fernández, and José A. López Brugos

University of Oviedo
Computer Science Departament. Campus de Viesques. Gijón. Asturias. 33204 Spain
lospinos@gmail.com, uo184703@uniovi.es, brugos@uniovi.es

Abstract. There are many tools that leans in the cognitivism as general platform, for example the wikis. This is a study in a topography course of Spanish students with no previous knowledge of computers and computing.We aplicated one voluntary work in their last project where it showed that the 59,4% of these students use in an effective manner the Wiki tool.

1 Introduction

Over the last decades, the efforts in teaching are becoming influenced by technological progress. Nowadays computer's are becoming more and more important in classrooms. Classes where educators and developers becomes agents of change, running these systems and making posible the methodological innovation. Actors that stablished the evolution of educational technology[1].

Web platforms solved some of the intercommunication needs requiered by users[2] . Their usability, their quickness, and their openness made posible the utilization of wikis in an educational environment[3].

In the next section the importance of wikis on education will be commented, and afterwards, the experiment realized with Topography spanish students will be described(at the University of Oviedo). The third section will be about results obtained and the methodology to get them. Finally the conclusions were wrote to share with the scientific community the strong and weak points found by the authors.

2 Wikis in Education

Although at Ward Cunningham [4] beginnings, at 1995's, created Wikis to a very different purpose from education (if encyclopedias are not an educative tool). Hopefully over time their use in educational field is appearing more frequently.

Internet's evolution trends produce what nowadays is called as Web 2.0 [5] technologies. It allows users to interact directly and have editable websites, in which modifications of a page can take only a few minutes, and becoming atractive to people because of simplicity [3].

On the last years, the use of wikis has been getting high. Indeed, their increasing numbers were compared with the virus behavioral growing [6]. So if

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 423–430, 2007.
© Springer-Verlag Berlin Heidelberg 2007

a years ago the needs about using multimedia, internet and videoconferencing on education was clear [7],[8] nowadays, they were gaining momentum again. Computational learning environments benefit from a strong background in educational theory [9].

These tools have changed the way of the students work. Wikis were employed to make easy getting knowledge by a collaborative work process[10]. Students can develop skills very useful for them like improving their intercommunications , analysis and synthesis abilities [11]. Such skills will be used in thoughts and information exchange [12].

Also a question to consider is the importance of teachers in the process because allthough using wikis on classrooms, they are the key anyway. They are needed to play the role of moderators and facilitators of contents[17], smothing the progress of students to be their own knowledge searchers. Wikis are webpages after all. So web metrics can be applied with them achieving a way to get more statistical information about the evolution of the learning. Web metrics are well document ways to reflecting user behaviour in a website [18]. In present a lot of web sites has them and the frequency of visits of a web portal is an indicator of their Internet's presence [19]. On the last years tools as Coremetrics [20], Web Side Story [21] or Click Tracks [22] are examples that are capable to accomplish that task.These kinds of tools help to achieve that task, because they provide to teachers with a source of up-to-time statistical data of assignments evolution.

2.1 Problem-Based Learning

Another learning system that suggests the increase of collaborative working was PBL (Problem Based Learning). It was a method based on the utilization of problems to acquire and integration of new concepts [23]. Although in its beginning was used for United States [24] and Canada's medicine courses[25], in present PBL was extended to other different disciplines and topics [26],[27]. PBL system of learning and teaching has several goals In example, the development of an attitude and critical judge, team work and independent reasoning and study habit [28]. It tries student identification of its learning needs [29]. That methodology creates a different skills acquisition way from teachers to student direct and passive transmission. Here in Spain, some developments about PBL and GIS [30].This translation to the Spanish cultural environment should be considered a very positive one in the GIS area, which is very linked with the topography students courses.

3 Methodologies

3.1 Methodology

The experiment was about the students exposition of a non Computer-Science bachelor to the CMS [31] developed by Wikimedia foundation [32]. The group chose were a first year university students from the "Mieres Technical Engineering School". Students were coursing their first year in a Topography degree.

The target was to use the Content Management System to let students create their own learning environment about a computer science introductory course. To prove the feasibility of our objective a proof case was developed.

The data obtained from the CMS, and also integrated with the course web page where processed by an Internet's statistics aggregator and marketing tool. This innovative use of the tool is also part of the success because it allows having real-time easy to process data from the experiment. It allows to visualized information from the enormous amount of information generated in the process.

3.2 Initial Conditions

Students had no previous knowledge of CMS, and their degrees of knowledge of computers vary from zero knowledge to average user knowledge. Students were part of a civil engineer degree. No other criteria were use to select them to be included on the learning study.

The groups of students were mainly men in an 60The classroom had 24 standard PCs with no special configuration and with no other technology to learn at the same time. The Content Management system chosen was Media Wiki. It is an open platform, well tested and documented (against million of users) with no fees cost and with a simple interface. All this characteristics allows students to "get the way" easily. And allow teachers to have a cheap software, easy to maintain tool that accomplish the resources' limitation environment that we had.

3.3 The Proof Case Implementation

The concept of a distributed CMS as the "Wikipedia" used is itself a complex one for novel students. So It was tested the comprehension of the mechanism by two ways. The test was developed as a voluntary work in their last project, and as a question in their final test (they were really interested in get the correct answer).

The voluntary work that was asked them to write had a 700 words' length, with hyperlinks and the addition at the end of the assignment their relevant bibliography. It was developed using the CMS of Wikimedia, the Media Wiki [32]. Only a very short explanation in the Media Wiki technology was gave. Students had to discover it by themselves.

Self-learning and guided learning are very different from the traditional method, but students adopted properly with no special difference from the old one.

The second part of the proof was adding a question linked with wiki in their final test exam. A very easy question indeed, any kind of simple test question about wiki should do the same effect for the proof case.

Both test presents a complementary approach to same question, can be learned the use of a wiki, and learn with that wiki over a first year topography course?

4 Results

Proof case shows that it's feasible from a grass root start to introduce the students to this technology in a quarter. It's at first glance at the question, a proof case, because students achieve to use it.

4.1 Assignments

The first part of our proof case was proposing an assignment to students. (1) was used to evaluate the average of three tasks completion:

$$[G] = (\frac{A + Ah + B}{3 * 10}) \quad (1)$$

Where: G = Grade, A = "Ability to write in CMS", Ah= "Ability to write hyperlinks", B = "Be able to represent their own bibliography".

The average result with this formula was 6.67 from people who answer something Here we also show statistics linked to the use of the contents developed by them from the CMS. Top ten references are showed on Table 1.

Some interesting result is also discovered here, not only contents linked with course were of interest of students. Some links created by them are not necessary of this kind of interest but also were visited. Learning dispersion should be expected in other e-learning environments. This is an open question for the

Table 1. Top ten visited wiki pages

Title	Clicks
Unicode	3.118
Main page	750
Interesting hyperlinks	718
Assignments	407
HTML	361
Google earth	208
Explain something	168
Access queries	150
GIS	94
Something	78

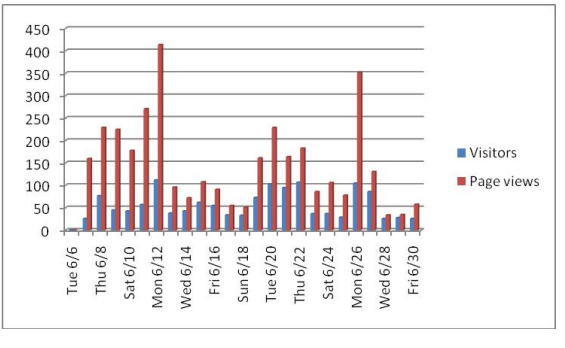


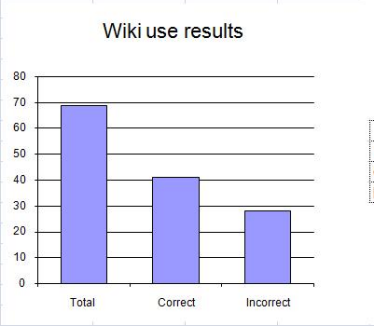
Fig. 1. Visitors and page views

future. But this studio reflects that technologies can help to learn, but also can help to distract students. It's a parallel and unexpected conclusion. Also to notice is that students followed a weekly period of activity. Period linked with their timetable, as you can see on Fig. 1.

4.2 Test's Question

As the second part of our proof case a question in the students final test was added. This question was really simple, and it was the kind of four possible answers question in which one response was: "A Wiki is a Content management system". But any kind of test's question will allow ask for the wiki comprehension.

The aggregated results appeared on Fig. 2.



	Aggregate
Total	69
Correct	41
Incorrect	28

Fig. 2. Test results graph

The graph shows that more students understand the concept than the ones that didn't understand it. Sixty nine people got the concept, in front only twenty seven didn't get it.

Of course, further work should be done to eliminate from our results the crossed factors as teacher's skills, classroom environment and others.

The test's results also show same conclusions as the first part of the proof case (assignments). Around 60% of the students achieve the objective fixed in our experiment.

5 Conclusions

Results were satisfactory on this experiment of using a wiki as a tool to support assignments on the first year course in the degree of Topography at the University of Oviedo. Some relevant experience was obtained. On teacher's side, making a wiki for the course was a innovative way that allow them to start a new planning process using wiki.

To students was also a motivating opportunity. It makes possible to them to know a new software tool, and at the same time learning and discovering new knowledge linked with their profession, and share it with their mates.

Related with the acquisition on domain's knowledge and wiki utilization, students get a 6'67 mark of 10 grades scale. Such a work was satisfactory considering the zero knowledge start point of some of the students. These students have done an big effort to get the hole idea.

On the second objective, the use of a wiki to knowledge transformation on our domain area, the 60% of positive answers were found. Their learning of the topic can be considered as good.

A positive side effect founded during the study's time was the increment of the number of students that regularly followed classes. The amount of increment was around 30%, but more precise measures should be done. It was also discovered an increase of cohesion between members of the same group while developing their assignment through the wiki. The last conclusion needs to be studied separately to isolate wiki effect from groups effects.

6 Future Work

Some ideas are now been explored by the authors:

First it's the need of a deeper statistical proof of method. With more groups, teachers and classes to test some of the new relevant facts founded.

It will be very new also to implement the same tasks with a semantic wiki. This new technology that combines a media wiki with semantic web concepts will make appears an entire new area of learning approaches using wikis because teachers will have an assistive technology to reasoning with students assignments and work in real time. Here the semantic interactions and conclusions obtained automatically will be a valuable source of information to explore.

Auto evaluation procedures are also a line to investigate, because wikis could simplify the possibility of help students to self-evaluate by comparison with teachers comments and with other students works.

Already under development, authors are working on a categories annotation system using media wiki technology to progressive evaluation and auto evaluation of students. This new use of wiki categories will be a way to reduce the time spent by teachers in the facilitation of information for student's evaluation in very big groups.

Afterwards, mobile technologies should be integrated and their interactions with wikis studied. Mobile technologies could give alerts and real-time info to students interested on topics events and classroom information.

The last question to explore is how to help students with special needs to use this kind of tools. Using assistive technologies also should help all the students to reduce time to learning in some topics, because accessibility and usability very often produce these effects when used in other areas of IT.

References

1. Ely, D.P.: Conditions that facilitate the implementation of educational technology innovation. Educational Technology 39(6), 23–27 (1999)
2. Schofield, J.: (May 6, 2003). Social Climbers. The Guardian. Retrieved from (March 6, 2004), Visited on (November 2006), http://www.guardian.co.uk/online/story/0,3605,950918,00.html
3. Report of the Information Services Working Group on Collaborative Tools. Collaborative Software Tools and Web 2.0. Version 6: (August 10, 2006)
4. Visited on (October 2007), http://c2.com/
5. O' Reilly, T.: What Is Web 2.0. Visited on (2006), http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html
6. Doebli, B.: Wikis - a Rapidly Growing Phenomenon in the German-Speaking School Community Pädagogische Hochschule Solothurn, Switzerland, Visited on (November 2006), http://www.wikipedia.com
7. Brugos, J.A.L.: Integrating videoconferencing in Intelligent Tutoring Systems. SCI'99: Third World Multiconference on Systemics, Cybernetics and Informatics, Orlando, Florida, EEUU (July 17-21 1999)
8. Brugos, J.A.L.: MITS: Foundation on Multimedia Instruction Tutoring Systems. SCI'98/ISAS'98: Word Conference on Systemics. In: Cybernetics and Informatics, Orlando, Florida, EEUU, vol. 3, pp. 379–386 (July 1998)
9. Henze, N., et al.: Adaptation in Open Corpus Hypermedia. International Journal of Artificial Intelligence in Education 12(4) (2001)
10. Visited on (December 2006), http://es.wikiversity.org/wiki/Portada
11. Worsham, D.: Making Wiki Work. Visited on (October 2006), http://www.seedwiki.com/wiki/
12. Reinhold, S.: WikiTrails: Augmenting Wiki Structure for Collaborative, Interdisciplinary Learning. In: Proceedings of the 2006 international symposium on Wikis, ISBN:1-59593-413-8
13. Visited on (December 2006), http://www.wikispaces.com/site/for/teachers/
14. Visited on (December 2006), http://www.edukalibre.org
15. Grierson, H., et al.: Supporting reflection and problembased Learning through the use of Laulima. Engineering and product design education conference, Napier University, Edinburgh, UK, (September 15-16, 2005)
16. Fuchs, F.: Wiki Communities in the Context of Work Processes. Visited on (November 2006), http://portal.acm.org/citation.cfm?id=1104973.1104977
17. Sanchez, J., et al.: Trabajo Colaborativo. Universidad de Chile. Gobierno de Chile. Ministerio de Educación. Visited on (October 2006) www.mmpchile.c5.cl/pag/press/Trabajo
18. Weischedel, B., et al.: Website optimization with web metrics: a case study. ACM International Conference Proceeding Series, vol. 156.
19. Díaz, H.: Medición de la Web, una escala de Interaccion parasocial para los portales World Wide Web. Revista Electronica de América Latina Especializada en Comunicación. Num 31
20. Visited on (October 2006), http://www.coremetrics.com/
21. Visited on (October 2006), http://www.websidestory.com/es/
22. Visited on (October 2006), http://www.clicktracks.com
23. Barrows, h.: Problem-based learning in medicine and beyond: a brief overview. In: en wilkerson, l., y gijselaers, w.h. (eds.) bringing problem-based learning to higher education: theory and practice, jossey- Bass publishers, San francisco (1996)

24. Visited on (October 2006), http://www.case.edu/
25. Visited on (October 2006), http://www.mcmaster.ca/
26. Visited on (October 2006), http://www.udel.edu/pbl/
27. Visited on (October 2006), http://pbl.stanford.edu/
28. Rojas, Y., et al.: Aprendizaje Basado en Problemas. Revista Facultad de Medicina. Visited on (October 2007), http://www.facmed.unam.mx/publica/revista/rev3-1999/aprendizaje.html
29. Branda, L.A.: Aprendizaje Basado en Problemas -Gua para el desarrollo de Recursos Humanos y Capacitacin Docente. Universidad de las Palmas de Gran Canaria. Visited on (October 2007)
30. Melero, C., Miguel.: Problems Based Learning, e-learning and GIS teaching
31. Boges Gouveia, J., et al.: Eftweb: Towards a Content Management System (2001)
32. Mediawiki Fundation, http://mediawiki.org

An Interactive Job Manager for Globus

Herbert Rosmanith, Dieter Kranzlmüller, and Jens Volkert

GUP, Joh. Kepler University Linz
Altenbergerstr. 69, A-4040 Linz, Austria/Europe

Abstract. These days, ongoing research towards interactivity to break the persevering batch processing paradigm in grid computing can be seen. Batch processing means: submit a job to a queue, process it and, only when the job is finished, analyse the results. By extending grid middleware through an *interactive jobmanager*, a novel approach to support applications which require interactive connections is provided, allowing event based processing of dynamic data created while a grid job is running. This allows users to utilise the grid for interactive applications much in the same way as on standard workstations and opens a series of new possibilities for next generation grid software.

1 Introduction

In most of todays grid computing environments, interactive jobs are rarely found. The main reason is the missing support for interactivity in grid middleware, so application programmers have to add interactivity at the application layer. Modifications in the middleware are extensive and nothing one would attack in order to support only one single application. Tasks of this magnitude are usually handled by teams, which work in projects specially formed for it.

- one cannot expect from an average grid programmer to modify the middleware - programmers want to use the grid, not to change it.
- modifications in the middleware need to be installed everywhere in the grid.

This can only be fulfilled within a project dedicated to middleware research. Modifications made and requested to be installed by grid programmers will rightfully fail to pass the administrative barriers.

In the rest of this work, we will focus on the Globus Toolkit[1]. Globus is the middleware which is most widespread used in grid computing. By implementing the solution for the Globus Toolkit, it is possible to reach most users in grid computing. Nevertheless we want to emphasise that the approach will in principle work with other middleware as well - the implementation will differ.

The obstacles mentioned above have been the reason for developing the standalone tool *glogin*[2], which provides interactive services without requiring modification of a single line of code of the middleware. However, weaknesses (such as stealing of computing time and data) in the middleware and in batch systems became evident, as has been shown in[4]. Additionally, this solution cannot be

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 431–442, 2007.
© Springer-Verlag Berlin Heidelberg 2007

integrated well (without additional changes) in the presence of a "resource broker", since the grid node, on which *glogin* will be executed, has to be known in the first place.

To overcome these problems, we have developed an interactive service as part of the middleware. During the design, value has been set on:

- Making modifications as small as possible
- Trying to build as additional software component instead of modifying the middleware core, so the resulting update will have less impact.

The resulting solution has been implemented for the "pre-web-services" environment. A web-services based implementation is in progress.

It is based on the analysis of how jobs are started in the grid. It was important to detect components which can be replaced as early as possible after job submission. It turned out that, in the Globus Toolkit, the "globus-job-manager" is the component in question.

This paper is organised as follows: Section 2 analyses interactivity in a Globus grid, Section ref:towards introduces the approach of the interactive jobmanager, while Section 4 describes the solution and implementation. Some examples on the usage of the interactive jobmanager are given in Section 5. An overview of related work is given in Section 6, before a summary and an outlook on future work concludes the paper.

2 Analysing Globus Toolkit 2

2.1 Data Transmission in the Grid

The standard jobmanager in Globus is the fork jobmanager (*jobmanager-fork*), which executes jobs on the computing element (CE). It makes use of two Perl scripts, which in turn use several Unix tools (*awk, grep, sed*). This leads to a delay of at least two seconds until the output of a program reaches the client which started the job. All further output of a running program will reach the client in a ten seconds interval, since in the *request* structure of the *jobmanager-fork*, the relevant field is set to a constant value.

Often, the *jobmanager-fork* is referred to as the "interactive jobmanager" (IJM). Certainly this is true to some extent, since the grid output can be transmitted (with delays) to the client while the program is running, if one does not care for the delay. Nevertheless a very important feature, which is crucial to interactive applications, is missing: the "feedback channel". This is an information channel which is required to be present in all interactive systems, even in non-technical, which allows passing back a response to an action to its initiator. The combination of forward and feedback channel forms an "Elementary Interaction Loop" (EIAL) [5]. In grid computing, the EIAL is expanded to include grid resources or even cross grid boundaries.

In [6], several attributes of interactivity are defined. One of these is "limited look-ahead", which means that the interaction partners do not know very well

what kind of response they have to expect from each other. This adds a "surprise value" to the messages being exchanged. With the Globus jobmanager, this quality of interactivity is limited. On the one hand, it is possible to define the input of a grid job by using the file IO ("stage-in") mechanism, on the other hand this mechanism restricts input to a set of predefined data. True interactive data, such as user input from devices like keyboard or mouse or readings from laboratory apparatus such as microscopes can not be sent to the grid job if they are generated while the job is executing. It is very likely that the "stage-in" mechanism tries to complete the data transmission at job initiation and does not honour new data during executing (a verification of this assumption is pending), so that a dynamic data path will have no effect, as one can consider examining the following invocation:

```
globusrun -w -r ce/jobmanager-fork '&
    (executable=/usr/bin/md5sum)(stdin=http://node/bigfile)'
```

2.2 Starting Jobs in the Grid

The ways the gatekeeper, jobmanager and GRAM-client communicate is shown in Figure 1. Thick continuous lines indicate HTTPS connections.

In the pre-web-services environment, a GRAM-client such as globusrun establishes a HTTPS-connection to the globus-gatekeeper, which usually listens for incoming connections on port 2119. After successful authorisation, the gatekeeper invokes the jobmanager. The socket descriptor of the conneciton is redirected to file descriptor 1, so the HTTPS communication can be done by using the "FILE *stdout" structure of the C language. The content of the request and the security context are passed in files which have been deleted by the gatekeeper, but still have an open file descriptor, so the content of the files, although possibly residing on the hard disk, is only available to the process holding the file descriptors. Interestingly, the number of the file descriptors is passed from the gatekeeper to the jobmanager using two environment variables, `GRID_SECURITY_HTTP_BODY_FD` and `GRID_SECURITY_CONTEXT_FD`.

After that, the jobmanager creates a "Job Contact", an HTTPS listener, which is used for passing data (such as "cancel", "status" or "signal" requests) from the client to the jobmanager [14]. The GRAM-client on its part uses the "Callback Contact", another HTTPS listener, used by the jobmanager for transmitting the jobstate of the grid job executing to the GRAM-client. It is interesting to note that, although the underlying TCP/IP protocol offers bidirectional communication, two such "contacts" need to be established, one for each direction. The reason is using HTTP for embedding the GRAM protocol [13]. The client/server architecture only allows sending HTTP requests from the client to the server, which the client responds to with an HTTP reply. A method for allowing the client and the server to exchange roles is not available in the HTTP standard.

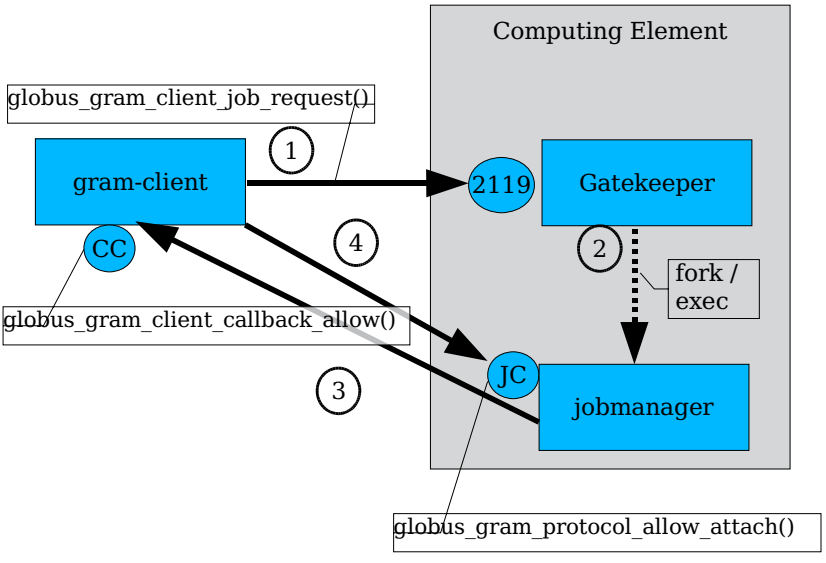


Fig. 1. Pre web-services job submission

Please note, for transmitting output data to the client, the GASS-Server can be used, which usually involves another HTTPS connection.

3 Towards Interactivity

After describing the operation of an existing approach, we will now address the qestion: how should an interactive jobmanager work? The central dogma of interactive programming is event based processing, which is focused on the user. Contrary to this works the polling mechanism of the batch oriented middleware. An interactive grid jobmanager has to operate on the events as they arrive and has to react as fast as possible. Ideally, this approach is not only applied for the feedback channel, but for other objects related to the grid job as well.

The question arises if one of the data paths in figure 1 can be used as interactive feedback channel. In this case the connection will be used in a different sense as originally intended, because on the receiving side, sent data will active the corresponding "callback routines", which are designed to handle the GRAM protocol. Reusing a data path would imply a change of protocol from a given point in time, such that an application specific protocol will then be used instead of the GRAM protocol. Therefore, the Job- and Callback Contact is out of question, since they are still in use while the grid job is running. Possibly the initial gatekeeper contact could be reused, but it turns out to be difficult for the client to find out the corresponding socket descriptor. This can only be accomplished by accessing internal data structures of the middleware, a step which we would like to avoid but becomes necessary anyway, as we will see later.

4 Solution and Implementation

Instead of reusing a connection by switching to a different protocol, we create a new TCP connection for dedicated use by interactive communication. Both forward and feedback channels are transported over this new bidirectional TCP connection. The connection can, but need not, be established by the GRAM-client starting the job: the advantage in this case would be that dynamically assigned port numbers can be used smoothly, which is not the case when using an "external" interaction client.

In any case, the connection parameters have to be passed to the grid middleware. The smartest way to acomplish this is to reuse existing methods. Parameters related to communication like GASS-URLs for stdin, stdout and stderr are passed via the Resource Specification Language (RSL), so the RSL is the most suitable approach to pass interaction parameters. Therefore, we expand the RSL and add another attribute, the *interactive* attribute, which has the following form:

```
(interactive=proto://host<:port>)
```

The interactive jobmanager uses this parameter to establish the interactive channel, once the GRAM connection establishment (see Figure 1) is completed.

The most challenging task when developing the interactive jobmanager was to identify the relevant parts in the code base of the middleware. Concerning Globus Toolkit 2 (GT2) version 2.4.3, the files in the globus-gram-jobmanager subdirectory amount for 13958, the files in globus-gram-protocol amount for 4002 lines of code. In contrast to this, the interactive jobmanager is only about 500 lines of code, which indicates that this program was developed for its limited task.

The interactive jobmanager only recognises a subset of the RSL. Attributes not supported will not lead to an error message but will be ignored. The following attributes are recognised:

```
executable
arguments
count
interactive
jobtype
directory
environment
```

Of particular interest are the *count* and *jobtype* attributes. As with the Globus jobmanager, *count* allows to invoke a particular number of executables. Legal values for *jobtype* are *multiple* and *mpi*. When starting jobs of type *multiple*, the jobmanager will invoke the respective binaries itself, when starting *mpi*-jobs, the MPI job starter *mpirun* will be invoked with the respective *-np* parameter.

The jobmanager communicates with the processes by using a "triple pipe" – a set of three unidirectional unnamed pipes, which realise the data transport of the standard file descriptors stdin, stdout and stderr. For each process, a triple pipe is allocated. In any case, data sent over the feedback channel will be forwarded to all processes. In the case of an MPI job, the only target for communication will be the *mpirun* process, since this is the only child process of the jobmanager. The *mpirun* process will forward its data to the "MPI master process" only (the process with MPI_Comm_rank=0). Data transmission to the other MPI processes will not be performed, which makes perfect sense, since the MPI jobs communicate with each other using the MPI protocol.

Monitoring of communication needs is done in the jobmanager from a central point. All relevant pipe descriptors and the socket descriptor related to the interaction channel are recorded in a "file descriptor set" `fd_set` and monitored by the `select(2)` system call. Only by using `select` is it possible to utilise event based processing and react to events as they arrive. This is the most important difference to the Globus jobmanager, which, at this point, will invoke a Perl module `fork.pm` in `$GLOBUS_LOCATION/lib/perl/Globus/GRAM/JobManager` to poll for changes every 10 seconds.

It is worth noting that processing GRAM events has to be utilised from the event processing loop, too. This is required to handle data from the "Job Contact" and trigger execution of registered callback routines. To solve this problem, it was necessary to access data structures internal to the grid middleware. The job contact is returned[1] to the jobmanager as a character string, forming an URL. The corresponding socket descriptor can be found in `globus_i_gram_protocol_listeners`. As the jobmanager notices IO activity on this socket descriptor, executing of callback routines is triggered by issuing a `globus_cond_wait`.

5 Examples

The way the interactive jobmanager works will now be illustrated with some examples. The hosts involved are:

clio, a gatekeeper
hydra, an interaction contact
medusa, a grid client

In all examples, *netcat* is used to create a TCP listener on *hydra*, which realises the interaction contact. *medusa* is the host the GRAM-client is started on and *clio* is the host which is configured as grid computing element, executing the Globus gsigatekeeper.

5.1 Counting and Delaying (Single and Multiple Jobtype)

The first example has already been used in demonstrating that jobmanager-fork delays output notably. When starting a program which increments a number

[1] From `globus_gram_protocol_allow_attach`.

(starting form zero), prints it and delays for one second, the output will arrive in a bulk every ten seconds. It can be shown[2] that this delay is generated at the grid side of the communication. A close inspection of Globus source code (e.g. using source code debuggers) reveals the hard coded polling interval (as mentioned above).

First, a TCP listener is started on *hydra*:

```
hr@hydra ~ $ nc -l -p 1234
```

Now, issuing a globusrun on *medusa*:

```
hr@medusa$ globusrun -w -r clio/jobmanager-inter '&
    (executable=count)(interactive=raw://hydra:1234)'
```

will invoke a program named *count* on *clio*. The interactive jobmanager, under whose surveillance the program is executed, will send the output to *hydra* on port 1234. We set aside reproducing the rather simple output, however, it is worth noting that the output at the client will show the same timing behaviour as the job executing on the grid.

To demonstrate multiple invocations of a program, a modification of *count* is used. For the sake of readability each process prepends its own process ID to the output lines. Using the RSL-attribute *count*, we specify invocation of five processes:

```
hr@medusa$ globusrun -w -r clio/jobmanager-inter '&
    (executable=count-pid)(count=5)(arguments=3)
    (interactive=raw://hydra:1234)'
```

On *hydra*, this will start five *count-pid* processes which count from zero to two. Each process will output a line containing its pid and the current number, delaying a second when counting. As above, the output will arrive at the client only delayed by network latency.

5.2 Using the Feedback Channel to Obtain Pseudo Terminals

Using *pty*, a stand-alone program which has specifically been developed for handling pseudo terminals, a login shell is started on *clio*. Terminal IO is redirected to *hydra*. An additional option *-R* has been introduced into *netcat*, which will put the local terminal (the one present on *hydra*, the host where *netcat* is executing) into "raw" data mode. This mode is used for character oriented IO, for transmission of control keys and to avoid local echoes.

On *medusa*, invocation of *pty* program on *clio* is issued by:

```
hr@medusa$ globusrun -w -r clio/jobmanager-inter '&
    (executable=src/pty)(interactive=raw://hydra:1234)'
```

[2] Using `my-gass-server, strace, tcpdump` or `netcat`.

In the terminal window on *hydra*, where *netcat* has been started, a login shell from *clio* will then appear:

```
hr@hydra ~ $ ./nc -R -l -p 1234
pseudo terminal on /dev/pts/58

excec .bashrc

hr@clio$
```

5.3 MPI Job with Feedback Channel

The next examples uses *mpi-pid-id*, a program which prints out its process ID and MPI_Comm_rank. After that, it works like the *cat* utility, but prepends its process ID to the output.

As an example, see the invocation on *medusa*:

```
hr@medusa$ globusrun -w -r clio/jobmanager-inter '&
    (executable=src/mpi-pid-id)(jobtype=mpi)(count=3)
    (interactive=raw://hydra:1234)'
```

All data is sent to *mpirun* and from there to the master process (and only to there), as the output shows (note that "foo" and "bar" is the input the user typed):

```
hr@hydra ~ $ ./nc -l -p 1234
pid=19845, id=1
pid=19846, id=2
pid=19844, id=0
foo
19844: foo
bar
19844: bar
```

5.4 Multiple Job with Feedback Channel

This example is similar to the one above, with the difference that the test program *pid-cat* does not contain MPI code. It shall clarify that data from the interaction channel is sent to all processes (user input is the word "foo"):

```
hr@medusa$ globusrun -w -r clio/jobmanager-inter '&
    (executable=src/pid-cat)(jobtype=multiple)(count=3)
    (interactive=raw://hydra:1234)'
```

Output:

```
hr@hydra ~ $ ./nc -l -p 1234
pid=20096
```

```
pid=20097
pid=20098
foo
20096: foo
20097: foo
20098: foo
```

Finally, we want to emphasise that using *netcat* is by no means obligatory, but used to demonstrate the communication flow in the examples. As has been mentioned above, the interaction channel can be utilised in the GRAM-client. This would, for example, result in clients like *globusrun-inter*. A minimalist version of this has been implemented.

6 Related Work

The fact that projects like the "Interactive European Grid Project"[15] (and its predecessor "Crossgrid"[16]) concentrate on interactive grids proofs that interactivity still is an important research topic. Interactive connections can be obtained in several ways, each having its advantages and disadvantages.

A common method is to allocate a "Network Virtual Terminal" (NVT), as described in the "Telnet Protocol Specification"[18]. This has been utilised by the telnet service and the remote shell (rsh) service. Being insecure, both have been replaced by the "secure shell" (SSH)[17]. In the grid environment, the SSH has been expanded to support grid related authentication. Within the Globus Toolkit 4 (GT4), the GSI enabled SSH is now available by default. The disadvantage of this approach is that a GSI aware ssh server (ssh daemon, sshd) needs to be installed, configured and activated on the target host. This conflicts with the approach that grid jobs are executed by sumitting them via the gsi-gatekeeper and results in access to the host without being subject to limitations such as grid job policies and scheduling. Therefore, GSI-SSH cannot be used as a replacement for the usual job submission mechanism.

Another solution which addresses interactivity for the grid is I-GASP [19]. The I-GASP system architecture uses "Virtual Network Computing" (VNC) [20] to display graphical output. This limits the useability of I-GASP in the area of grid based visualisation, since VNC is not designed for video compression. With the interactive jobmanager, VNC can be supported too, but there is no limitation to VNC as with I-GASP. In contrast, the interactive jobmanager can support true grid based visualisation such as Gvid [21] easily. Additionally, I-GASP features a "Grid Interactive Shell" (GISH), which shall protect against malicious users by allowing only a subset of commands. Even programs compiled by a grid user are inspected at the source code level and certified if considered safe. Unfortunately, the authors of I-GASP fail to show how such as correctness verifier has been implemented.

7 Conclusion and Future Work

The interactive jobmanager fulfills the limited task of supporting interactive applications and thus is relatively small compared to the Globus GRAM jobmanager. Therefore it is an attractive option to add its functionality to the globus jobmanager, allthough this contradicts with the initial postulation to provide the software as a seperate module. The bonus of this approach is that reimplementation or copying of functionality present in the globus jobmanager (such as staging of files) can be avoided. The basic idea is adding the interactive contact to the jobmanager's *request* structure and providing the necessary callback routines.

Further research will be necessary to see how to combine event based and polling based processing in a single programm. Although the straightforward idea looks simple, again, `select` offers all the requirements needed in a single system call, which is: notification of IO events and notification of timeout. IO events lead to a state change in the file descriptor set, thus event based processing has to be called. Detecting a timeout requires extra work, if it should be implemented in a portable way. Some operating systems modify the timeout field to indicate the time left, others do not[3]. In case of a timeout, polling based processing has to be called.

The current implementation does not evaluate the protocol parameter of the interaction contact. Although in the examples "raw" is being used, this value is ignored. The protocol parameter could be used to enable authentication and encryption, for example, by specifying "gss://host:port", usage of the GSS-API[9] would be activated. Furthermore, by adopting the GSH-protocol[3], TCP and X11 forwarding can be provided by the jobmanager, thus reducing communications overhead as compared to providing the same functionality by stand-alone tools. On the other hand, this increases the complexity not only of the jobmanager, but of the client as well, since the client has to understand the GSH-protocol too.

Allthough only *multiple* and MPI jobs are currently supported, it could also be interesting to support local resource managment systems such as the Portable Batch System (PBS) [8] or Condor [7], by adopting the Posix Standard 1003.2d [10] for distributed batch queuing environment.

The most crucial point is, that for ease of development, the jobmanager has been created for Globus Toolkit 2. The current Globus release is the Globus Toolkit 4 (GT4), which uses web-services. Therefore, porting the interactive jobmanager to GT4 is currently underway. However, programming with web-services (WS-GRAM) is much more complicated than with pre-web-services GRAM.

Finally, exploring other grid middleware such as gLite[11] and Unicore[12] in order to add support for interactivity in the way described in this paper is required to be done in the long run.

Acknowledgments

The work described in this paper was supported in part by the European Union thr ough the FP6-2005-Infrastructures-7- contract No 031857 project "Int.eu.grid".

[3] Unfortunately, POSIX is ambigous and permits either behaviour!

References

1. Foster, I., Kesselman, C.: Globus: A metacomputing infrastructure toolkit. The International Journal of Supercomputer Applications and High Performance Computing 11(2), 115–128 (1997)
2. Rosmanith, H., Volkert, J.: glogin - Interactive Connectivity for the Grid, Distributed and Parallel Systems: Cluster and Grid Computing DAPSYS, Austrian-Hungarian Workshop on Distributed and Parallel Systems, p3-12 (2004)
3. Rosmanith, H., Volkert, H.: Traffic Forwarding with GSH/GLOGIN., 13th Euromicro Workshop on Parallel, Distributed and Network-Based Processing PDP, pp. 213–219 (2005)
4. Rosmanith, H., Praxmarer, P., Kranzlmüller, D., Volkert, J.: Towards Job Accounting in Existing Resource Schedulers: Weaknesses and Improvements. In: Gerndt, M., Kranzlmüller, D. (eds.) HPCC 2006. LNCS, vol. 4208, Springer, Heidelberg (2006)
5. Cronholm, S.: Need for Action Oriented Design and Evaluation of Information Systems. Human - Computer Interaction, Theory and Practice (Part I), 306–310
6. Brand, S.: The Media Lab: Inventing the Future at M.I.T, Penguin, pp. 46 (1988)
7. Litzkow, M., Livny, M., Mutka, M.: Condor - a hunter of idle workstations. In: Proceedings of the 8th International Conference of Distributed Computing Systems, pp. 104–111 (1988)
8. TORQUE Resource Manager, `http://www.clusterresources.com/pages/products/torque-resource-manager.php`
9. Linn, J.: Generic Security Service Application Program Interface, RFC 2743, Internet Engineering Task Force (January 2000)
10. IEEE Std, 1003. 2d-1994 IEEE Standard for Information Technology Portable Operating System Interface (POSIX) – Part 2: Shell and Utilities – Amendment 1: Batch Environment -Description
11. gLite: Lightweight Middleware for Grid Computing, `http://glite.web.cern.ch/glite`
12. Erwin, D., Snelling, D.: UNICORE: A Grid Computing Environment. LNCS (2001)
13. Czajkowski, K., Foster, I., Karonis, N., Kesselman, C., Martin, S., Smith, W., Tuecke, S.: A Resource Managment Architecture for Metacomputing Services. In: Proceedings of IPPS/SPDP '98, Workshop on Job Scheduling Strategies for Parallel Processing, pp. 62–82 (1998)
14. GRAM Protocol Definition, `http://www-unix.globus.org/api/c/globus_gram_protocol/html`
15. Interactive European Grid Project, `http://www.interactive-grid.eu`
16. The Crossgrid Project, `http://www.crossgrid.org`
17. Ylönen, Tatu. SSH Secure Login Connections over the Internet, Sixth USENIX Security Symposium, of the Proceedings, SSH Communications Security Ltd. pp. 37-42 (1996), `http://www.usenix.org/publications/library/proceedings/sec96/full_papers/ylonen/`
18. Postel, J., Reynolds, J.: Telnet Protocol Specification, RFC854, Internet Engineering Task Force (May 1983)
19. Basu, S., Talwar, V., Agarwalla, B., Kumar, R.: Interactive Grid Architecture for Application Service Providers, Mobile and Media Systems Laboratory, HP Laboratories Palo Alto, Technical Report (July 2003)

20. Richardson, T., Stafford-Fraser, Q., Wood, K., Hopper, A.: Virtual Network Computing. IEEE Internet Computing 2(1), 33–38 (1998)
21. Köckerbauer, T., Polak, M., Stütz, T., Uhl, A.: GVid - video coding and encryption for advanced Grid visualization. In: Volkert, J., Fahringer, T., Kranzlmüller, D., Schreiner, W. (eds.) Proceedings of the 1st Austrian Grid Symposium, vol. 210 of books@ocg.at, Schloss Hagenberg, Austria, Austrian Computer Society, pp. 204–218 (2006)

Interactive Molecular Dynamics Simulations on the Grid

Rene Kobler[1], Thomas Köckerbauer[1], Ulrich Omasits[2], Martin Neumann[2], Wolfgang Schreiner[2], and Jens Volkert[1]

[1] GUP, Joh. Kepler University Linz,
Altenbergerstr. 69, A-4040 Linz, Austria/Europe
rkobler@gup.uni-linz.ac.at
http://www.gup.uni-linz.ac.at/
[2] MSI, Medical University of Vienna
Spitalgasse 23, A-1090 Vienna, Austria/Europe

Abstract. The impressive growth of life sciences in the last couple of years forced computer scientists to deal with scientific areas like biochemistry, biophysics or medicine. Studies in these research areas are often based on dynamic molecular systems whose principles are found in basic mechanics, forming the term "molecular mechanics". Since such highly computationally intensive molecular dynamics (MD) simulations [12] are usually carried out through the variation of different input parameters, they are well suited for being executed in heterogenous grid environments. This paper introduces a novel technique for interactively visualizing MD simulations on the grid, a method currently not available for grid-based MD simulations, supporting physicians, chemists and biophysicists of being able to abort a simulation when it turns out that it would not make sense to run it even longer.

1 Introduction

Most today's grid [3] applications are limited in terms of interactivity, that is the behavior of such applications is quite similar to that of mainframe batch applications a couple of decades ago. Grid jobs are submitted to a queue which is usually responsible for an intelligent distribution to the available grid resources, the results are then post-mortem analyzed by the scientist. Several grid middlewares apply this technique (i.e. Globus Toolkit [4]), the scientist is pensalized in a way that he has to wait until the whole job is being finished.

glogin [13] addresses this lack within Globus Toolkit, it offers the possibility for tunneling arbitrary traffic into and out of the grid. The users are able to apply different interactive services, i.e. low-level communication services for visualization, grid pseudo-terminal functionality and shell, TCP forwarding in and out of the grid, X11 remote displays for grid application as well as virtual private networks inside and outside the grid. These characteristics emerges to be useful when the observation of long running grid applications is desired.

MD simulations are expected to be one of the most challenging high performance computing applications in nowadays computer science. The N-body

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 443–447, 2007.
© Springer-Verlag Berlin Heidelberg 2007

problem which originates in the dynamics of the solar system is central to understanding matter at the microscopic level. And it is the task of the numerical solution of this problem that MD addresses [12]. The theoretical basis for MD is to be found in analytical mechanics, i.e. Newton and Euler revolutionized this field of natural sciences. Newton's second law of motion can be considered to be the fundament of the simplest form of MD. Depending on the model complexity run times of several hours up to days are no rarity.

When aiming to run MD simulations on the grid, it is sometimes of tremendous interest to get some feedback, recognizing if it still makes sense to finish the simulation regularly. This facility is especially helpful when analyzing the binding properties of genomes to proteins in the field of immunology. The basic principles of the underlying medical studies deal with the human immune system whose understanding is necessary for research in the fields allery, diabetes, multiple sclerosis, cancer and so on. The theoretical background for this work focuses on the human immune system which detects and destroys pathogenic invaders, parts of which are held in special MHC (Major Histocompatibility Complex) molecules as 'epitopes'. It is simulated how epitopes are held in order to make predictions for the development of vaccines [8]. Figure 1 illustrates the processes of failed binding by the corresponding MHC molecule. A MD system is applied on the model to be simulated, the results of the molecular dynamics simulation are usually stored in trajectory files, which can be visualized under the usage of different trajectory visualization toolkits.

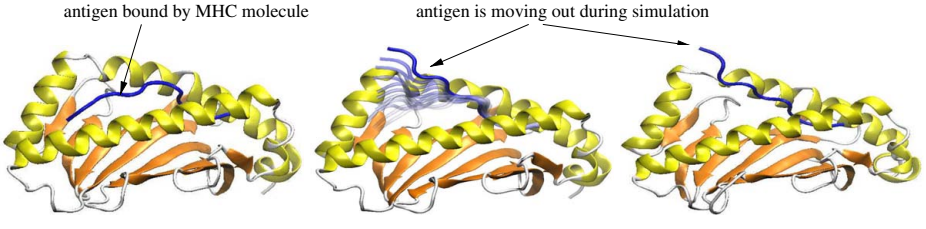


Fig. 1. Failed binding during MD simulation. The left part illustrates the starting point while the middle part shows the escape of the antigen resulting in the situation characterized by the right part.

In our studies we basically applied Gromacs [1] as MD simulation package, because it was easy to configure and install at all of our Austrian Grid [14] resources.

Since a simulation time of 2 - 3 days (using 8 processors on an SGI Altix 350) is rather high compared to a simulated time of 1 ns, it is desireable that the simulation of unsatisfactory parameterized models is recognized as soon as possible (see Figure 1). One conceivable solution could be to offer the scientist a possibility to examine the simulated system during runtime. To achieve this glogin has to be integrated to access the grid application during runtime. Its main task is to transfer the output of the preferred visualization toolkit in a reasonable

time to the scientist. Finally the scientist is able to cancel the simulation when it seems appropriate.

The visualization of the trajectory files is accomplished in a best way using VMD (Visualizing Molecular Dynamics) [6]. Since VMD is able to view a running simulation, we had to take the transfer time of visualization data to the client into consideration. Grid middlewares offer secure communication due to encryption of data, therefore a method has to be applied reducing the transfered data while still keeping the quality of the results. Within our system GVid [9] is responsible for transporting the visualization output in form of a video stream to the user client, the current implementation of GVid is based on XVid [15] which is an implementation of a part of the MPEG 4 standard. To enable a binding of GVid to VMD we had to extend VMD with a new GUI Toolkit implementation, that is GLUT, besides the already existing ones.

In the first implemented prototype of the above shortly described scenario MD simulations are performed by Gromacs. Unfortunately it is not possible to bind Gromacs to VMD, so that an on-line visualization of the simulated system results. But Gromacs is well suited for a proof of concept without adapting the current simulation configuration to an other MD simulation package.

The next section gives a short overview of other work related to this regarding interactivity. Finally an overview of the demonstrated system is given.

2 Related Work

Different grid projects funded by the European Union deal with interactivity in grids. The EU *CrossGrid* project [2] focused on compute- and data-intensive applications which requires an interaction with a person in a processing loop. Within the scope of CrossGrid, the Grid Visualization Kernel (GVK) [5] was developed and extended. GVK splits a visualization pipeline, to reduce computational demands on client side. Generally spoken, an application produces simulation data, a visualization mapping process generates "Abstract Visualization Objects", so called intermediate geometry data, which are transformed into a displayable image by the rendering process. It turned out, that the processing speed increases when shifting the image generation process to the grid, but a bottle-neck was still existent due to transmission to the client.

The *int.eu.grid* which started at the end of year 2006 tries to solve this problem. The core of this project is an "Fusion Plasma Application" which produces an interactive visualization of a distributed simulation executing on the grid. Like the above described scenario this project harnesses glogin as well as GVid, the application had to be redesigned, splitting it into a visualization client, and into several MPI jobs, responsible for the computing tasks. Additionally the "Migrating Desktop" [11] and the "Roaming Access Server" [10] are applied and modified within this core application.

Generally, many grid portals offer interactivty but not in that way we understand interactivity. The term interactivity mostly refers to an interactive way of using the grid portal.

3 System Overview

Figure 2 depicts our test system, strictly splitted into grid part as well as user interface part. The user first has to submit the MD simulation job into the grid, due to the necessary docking to VMD the simulation is carried out using NAMD [7]. GVid is splitted into two parts, a grid part located on the node which takes on the rendering task responsible for enocding of frames and on the other hand a UI part for frame decoding. Additionally GVid is also able to receive user events at client side, encode it, transport it to GVid's binding to VMD, decode it and pass it to VMD for further processing, so full steering of VMD is made possible. The window accentuated in the upper right part of the figure simply shows an MPEG 4 encoded output of VMD.

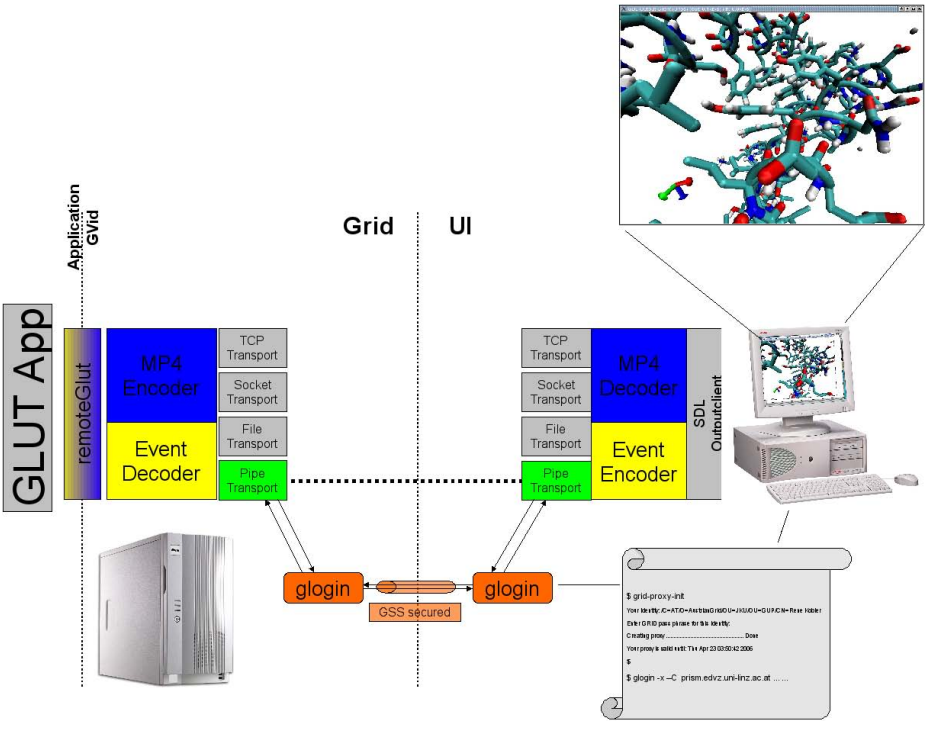


Fig. 2. Visualizing running grid MD simulations

4 Conclusions and Future Work

In combination with glogin, GVid is an interesting technique for transporting complex visualization outputs over the grid. This especially turns out to be useful, when rendering tasks require powerful graphics hardware, which is often not available on standard grid portals. Furthermore the proposed system enables

the scientist to obtain a fast replay of the MD simulation, framerates of about 20 fps have been achieved when transporting encoded video data from Linz (Austria) to Dresden (Germany).

Acknowledgements. The work described in this paper is partially supported by the Austrian Grid Project, funded by the Austrian BMBWK (Federal Ministry for Education, Science and Culture) under contract GZ 4003/2-VI/4c/2004.

References

1. Berendsen, H.J.C., van der Spoel, D., van Drunen, R.: GROMACS: A message-passing parallel molecular dynamics implementation. Computer Physics Communications 91, 43–56 (1995)
2. Bubak, M., Malawski, M., Zajac, K.: The CrossGrid Architecture: Applications, Tools, and Grid Services. In: Proc. of the 1. European Across Grids Conference, Santiago de Compostela, Spain (2003)
3. Foster, I., Kesselman, C. (eds.): The Grid: Blueprint for a New Computing Infrastructure. Morgan-Kaufman, San Francisco (1999)
4. Foster, I., Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit. The International Journal of Supercomputer Applications and High Performance Computing 11(2) (1997)
5. Heinzlreiter, P., Kranzlmüller, D.: Visualization Services on the Grid: The Grid Visualization Kernel. Parallel Processing Letters 13(2), 135–148 (2003)
6. Humphrey, W., Dalke, A., Schulten, K.: VMD - Visual Molecular Dynamics. Journal of Molecular Graphics 14(1), 33–38 (1996)
7. Kale, L., et al.: NAMD2: Greater scalability for parallel molecular dynamics. Journal of Computational Physics 151, 283–312 (1999)
8. Kobler, R., et al.: Computer Simulation of T-Cell Adaptive Immune Response Utilizing the Austrian Grid Infrastructure. In: Volkert, J., Fahringer, T., Kranzlmüller, D., Schreiner, W. (eds.) 1st Austrian Grid Symposium, Schloss Hagenberg, Austria, Schriftenreihe OCG, Band 210 (May 2006)
9. Köckerbauer, T., Polak, M., Stütz, T., Uhl, A.: GVid - Video Coding and Encryption for Advanced Grid Visualization. In: Volkert, J., Fahringer, T., Kranzlmüller, D., Schreiner, W. (eds.) 1st Austrian Grid Symposium, Schloss Hagenberg, Austria, Schriftenreihe OCG, Band 210 (May 2006)
10. Kupczyk, M., Lichwala, R., Meyer, N., Palak, B., Plociennik, M., Wolniewicz, P.: Roaming Access and Migrating Desktop. In: Proc. of the Cracow'02 Grid Workshop, Cracow, Poland (December 2002)
11. Kupczyk, M., Lichwala, R., Meyer, N., Palak, B., Plociennik, M., Stroiski, M., Wolniewicz, P.: The Migrating Desktop as a GUI Framework for the Applications on Demand. In: Bubak, M., et al. (eds.) ICCS 2004. LNCS, vol. 3036, pp. 91–98. Springer, Heidelberg (2004)
12. Rapaport, D.C.: The Art of Molecular Dynamics Simulation, 2nd edn. Cambridge University Press, Cambridge (2004)
13. Rosmanith, R., Kranzlmüller, D.: glogin - A Multifunctional, Interactive Tunnel into the Grid. In: Proc. Grid 2004, 5th IEEE/ACM Intl. Workshop on Grid Computing, pp. 266–272. IEEE Computer Society, Pittsburgh, PA, USA (2004)
14. The Austrian Grid Initiative, `http://www.austriangrid.at`
15. The Xvid project, `http://www.xvid.org`

Extending the Globus Architecture with Role-Based Trust Management*

Maurizio Colombo, Fabio Martinelli, Paolo Mori, and Anna Vaccarelli

Istituto di Informatica e Telematica
Consiglio Nazionale delle Ricerche
Via Moruzzi 1, Pisa 56124, Italy
{maurizio.colombo,fabio.martinelli,paolo.mori,anna.vaccarelli}@iit.cnr.it

Abstract. Grid technology concerns the sharing of resources among a very large set of users. One of the main security issues of the Grid environment concerns the user authorization. As a matter of fact, Grid resource providers grant accesses to their resources to possibly unknown Grid users, but they want that these accesses are regulated by proper security policies.

This paper proposes a framework that integrates an advanced authorization system, the RTML one, in the Globus toolkit. For each Grid user that requests to access the Grid resource, the framework determines the proper set of rights to be paired to the user depending on the trust he previously collected interacting with other sites on the Grid, instead of simply considering his identity. This trust is represented by the set of credentials issued by other Grid sites that grant to the user some roles in these sites.

1 Introduction

Grid technology supports the creation of set of users, denoted as Virtual Organizations (VOs), that share their resources. Security is a fundamental issue in the Grid, due to the dynamic, collaborative and distributed nature of this environment. As a matter of fact, the resource sharing supported by the Grid must be highly controlled, because Grid service providers, that would grant access to their resources to other Grid users, want these accesses to be regulated by a security policy stating which actions can be performed on their resources.

Security management in the Grid environment is complicated by the need to establish secure relationships between a large number of dynamically created participants among which no direct trust relationships may exist a priori, and across distinct administrative domains, each with its distinct resources and its own local security policy. In the case of computational resources, for instance, Grid service providers allow the execution of unknown applications on behalf of unknown Grid users on their resources. If an adequate security support is not

* Work partially supported by EU-funded project *Trust and Security for Next Generation Grids*, GridTrust, IST-033817.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 448–456, 2007.
© Springer-Verlag Berlin Heidelberg 2007

adopted, the applications could perform dangerous and even malicious actions on these resources. Hence, among the functionality of Grid security support, the authorization is a fundamental one.

1.1 Paper Contribution

The contribution of this paper is the enhancement of the security infrastructure of Globus [2], a widely used Grid toolkit, through the integration of a powerful authorization service, the RTML Framework [7]. The Globus Toolkit provides a basic authorization system, the gridmap one, that is based on the static configuration of a file, the gridmap file. In this file, the identity of each authorized user is statically mapped on a local account that is exploited to execute remote jobs on behalf of the user. Hence, given a Grid resource R, only users that have been previously registered in the gridmap file of R can access the services provided by R. This feature could be a limitation in an open and distributed environment such the Grid one.

Alternative solutions have been proposed to improve the authorization service of Globus, as described in Section 5. However, these solutions do not take into account the trust that the Grid user collected in his previous interactions with Grid resources.

The authorization system we propose, instead, determines the rights of a Grid user on a Grid resource taking into account the complex trust relationships he collected in the past by exploiting services on other Grid resources. These relationships are expressed in form of credentials, issued by these Grid sites. From the Grid resource point of view, the credentials owned by a Grid user determine the role of this user on the resource. Each role corresponds to a set of privileges, and is enforced through local accounts with the corresponding set of rights. These accounts are dynamically paired to Grid users' identity by our framework that automatically updates the gridmap file. This implies that, when a Grid user accesses a Grid resource, its identity is verified in the standard way, but it is not required that its name has been previously registered on the resource. Hence, the local account in which is mapped a Grid user does not depend on his identity, but on the credentials he owns. Obviously, distinct users are mapped onto distinct local accounts. Moreover, this set of credentials is dynamic, because new credentials could be added by other Grid sites and some of the existing ones could be expired. This implies that two requests submitted by the same Grid user in distinct times, could be mapped on distinct local accounts on a Grid resource if the set of credentials is changed in the meanwhile.

1.2 Paper Structure

The paper structure is the following. Section 2 briefly describes the RTML framework, while Section 3 describes the integration of this framework in the Globus Toolkit. Section 4 presents an example of access rules and user credentials that could be evaluated by the RTML authorization system in a real job request.

Section 5 describes some previous attempts to improve the Globus Security Infrastructure by integrating some authorization systems. Finally, Section 6 concludes the papers and presents some future work.

2 Role-Based Trust Management Framework

The *RT Role-based Trust-management* framework provides policy language, semantics, deduction engine, and concrete tools to manage access control and authorization in large-scale and decentralized system. *RT* combines the strength of Role-Based Access Control (RBAC) and Trust-Management (TM). RBAC was developed to manage access control in a single organization in which the control of role membership and role permissions is relatively centralized in a few users. *RT* takes from *RBAC* the notion of role to assign permissions to users. TM is an approach to distributed access control and authorization in which access control decision are taken on the base of *policy statements* made by multiple principals, that in the case of interest will be Grid sites. From TM, *RT* takes the principles of managing distributed authority through the use of credentials, as well as some notation denoting relationships between those authorities. The main concept in *RT* is the notion of *roles*: each *RT* principal has its own name space for defining roles, and each role is compounded by the principal name and a *role term*. For example, if K_A is a principal and R is a role term, then $K_A.R$ denotes the role R defined by principal K_A. Only K_A has the authority to issue policy statements defining the members of the role $K_A.R$.

A detailed description of the RT framework can be found in [5,8,9,15].

3 Globus Extended with RTML: RTMLAuthzService

This section describes the integration of *RTMLAuthzService*, an authorization system based on the RTML framework, within the Globus Toolkit. The RTMLAuthzService determines the permissions to be paired to a Grid user depending on the trust that he collected in his previous experiences on the Grid. In particular, RTMLAuthzService was developed for those services that require the use of a local account to execute the user request, such as the *ManagedJobFactoryService*, that is the Globus service that shares the computational resource. The RTMLAuthzService provides the access to the resource only basing his decision on the roles granted to the user according to his own credentials. The architecture for the resulting system is shown in Figure 1, and is described in the following of this section.

The Grid user submits his request for a given service to the Globus container. The Globus container evaluates the requests and, to determine the user rights on the requested service, it executes the authorization module that is indicated in a configuration file (server-config.wsdd) paired with this service. This file is configured to exploit the RTMLAuthzService, through the *SamlAuthorizationCallout* mechanism [11]. This mechanism allows the Globus container to exploit external authorization systems, that run as services in the container itself. The

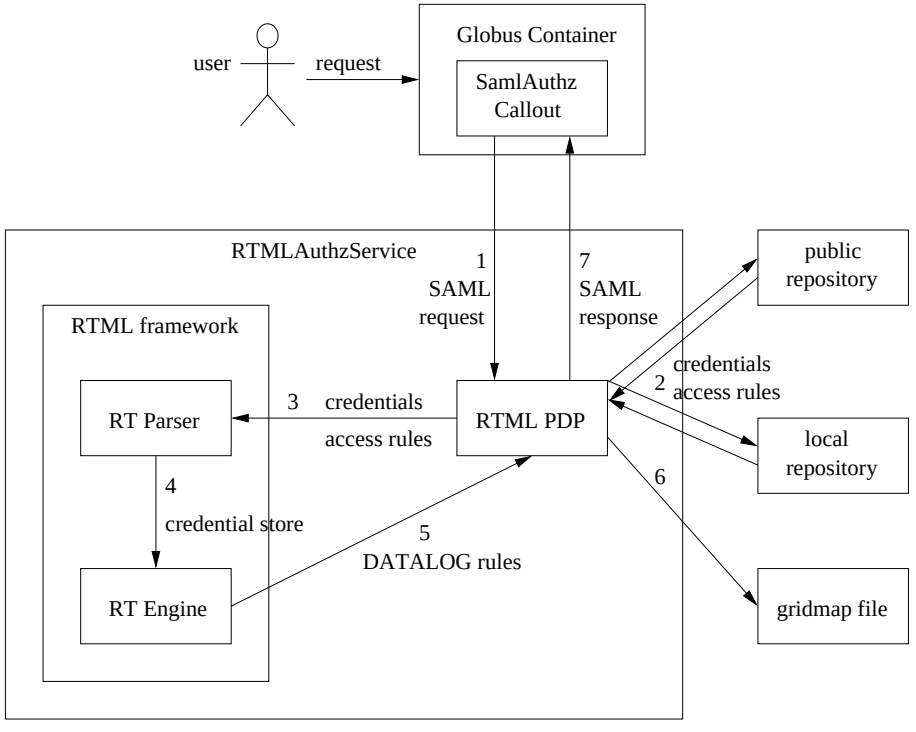


Fig. 1. The RTMLAuthzService system

communication between the SamlAuthorizationCallout module and the RTMLAuthzService exploits the *Saml* protocol. A SamlRequest is generated by the SamlAuthorizationCallout module and it is sent to the RTMLAuthzService, as described by the arrow with tag **1** in Figure 1. This request contains information about the user, the service requested and the credential repositories. The RTML Policy Decision Point (RTML PDP) determines the rights to access the service. If the access is granted, the Globus container receives a positive SamlResponse and starts the execution of the requested service, otherwise it receives a negative SamlResponse (represented by the arrow with tag **7** in Figure 1).

3.1 RTML Policy Decision Point

Once the request has been acquired, the RTMLAuthzService needs to evaluate the credentials that grant to the Grid user specific rights on the requested service, according with the policies. The RTML PDP is the component to do that. To retrieve the credentials of the user, two different approaches are possible: the *push model*, in which the user chooses the most suitable subset of credentials to submit, and *pull model*, in which is the RTML PDP that obtains the user credentials from repositories whose URIs were passed at initialization time. Even though the former could be better concerning the user privacy, because a user

does not need to disclose all his credentials, for a first implementation of the RTMLAuthzService we choose the *pull model* in order to keep unchanged the Globus Toolkit. Hence, the RTML PDP downloads the user credentials from the public repositories, and other credentials (e.g. the site access rules) are retrieved from a local repository (as shown by arrow **2** in Figure 1). Then, the RTML PDP checks the credentials integrity verifying their signature, and passes the related XML code to the RTML Framework parser (as shown by arrow **3** in Figure 1). As a matter of fact, credentials and access rules that have been issued by other entities must be signed. The RTML Framework transforms the credentials into DATALOG statements and returns them to the RTML PDP. The RTML PDP takes its access decisions depending on the action requested and the service to be accessed, exploiting the user credentials expressed by the DATALOG statements. In particular, a configuration file defines which roles are required to execute a given action on a given service, and the RTML PDP evaluates the DATALOG statements through a Prolog engine to determine which of these roles could be granted to the user.

3.2 RTML Framework

The RTML framework consists of the *RT Parser* and the *RT Engine*. The *RT parser* is the element that receives the XML code representing the credentials from the RTML PDP and that parses it. Its output is a complex data structure, a *CredentialStore*, which includes all the information concerning these credentials and all the site's access rules in the form of *RT statements* (Figure 1, arrow **4**).

A second step is done by the *RT engine* which converts each *RT* statement in a *DATALOG* statement according to the rules and the specific syntax explained in [7]. These statements are returned to the RTML PDP for the enforcement (arrow **5** in Figure 1).

3.3 Gridmap File

The Globus standard authorization service exploits the static configuration of the gridmap file to map the Grid user identity onto a local account. This represents a limitation in the authorization system because it makes the system closed. As a matter of fact, an entry in the file constitutes a necessary condition for a user who wants to access a service.

For a matter of compatibility with the Globus Toolkit, each time the RTMLAuthzService grants to a user the permission to access a service, it maps the Grid user identity on a local account, that is dynamically created according to the rights that are paired to the role, and adds a proper entry to the gridmap file. Sometimes the RTML PDP grants more than one role to the user. If the user specifies the role he wants in the *JobDescriptionFile* and this role is granted, the grid user will execute the job with the privileges granted by this role. If no role is specified, the role with the minimum privilege is used.

4 Example

Let us suppose that the Center of Electronic Computation (CCE) laboratories offer computational services to the university students which are carrying out a stage at ABC company. To access the computational service, a user need to supply a credential issued by the ABC company, that asserts that he is doing a stage, and a credential issued by a university, that grants him the role of student. Furthermore a credentials chain could be required to verify that this university is one of the admitted by the CCE laboratories. In this situation the user identity is irrelevant in the evaluation of the request, while the roles he owns in the involved domains are decisive. Hence, each university student that owns the credentials can access the CCE laboratories computational service, even if his identity has not been registered by the ABC site. Let us suppose that Alice, that is a student of XYZ University and is carrying out a stage at the ABC company, wants to access the CCE computational service, and that the following are the credentials held by the Alice:

1. *XYZ.Student(university='University of XYZ', department='Informatic', id='1999s131', firstname='alice', lastname='rossi')* $\leftarrow$ *Alice*
2. *ABC.Collaborator(role='stage', number='IS137', firstname='alice', lastname='rossi')* $\leftarrow$ *Alice*
3. *MIUR.University(name='University of XYZ')* $\leftarrow$ *XYZ*

The Access Rules stored by CCE are the following:

1. *CCE.University(name=?)* $\Leftarrow$ *MIUR*
2. *CCE.Student(university=*ref_{uni}*, department=?, id=?, firstname=?, lastname=?)* $\Leftarrow$ *University(name=*ref_{uni}*)*
3. *CCE.ABCGuest* $\leftarrow$ *ABC.Collaborator(role=?, number=?,firstname=*ref_{first}*, lastname=*ref_{last}*)* $\cap$ *CCE.Student(university=?, department='Informatic', id=?, firstname=*ref_{first}*, lastname=*ref_{last}*)*
4. *CCE.Guest* $\leftarrow$ *CCE.Student(university=?, department='Informatic', id=?, firstname=?, lastname=?)*

In the previous credentials the symbol '?' is used to denote a parameter whose value is not specified and $\cap$ is the conjunction of constraints, as explained in [6]. Alice represents the DN of the principal specified in the first two credentials for the roles issued by the entities whose DNs are XYZ and ABC. The owner of the third credential is XYZ University and not Alice. This statement is needed because it could be used to infer information about Alice's roles. As a matter of fact, ABC consider as University the principals that are considered Universities by MIUR.

These credentials and access rules are retrieved by the RTML PDP from the repositories, and are passed to the RT Engine. The RT Engine transforms the previous credentials and access rules in the following set of DATALOG statements:

1. *Student(XYZ, Alice ,* x_1, x_2, x_3, x_4, x_5*)* :- x_1 = *'University of XYZ'*, x_2 = *'Informatic'*, x_3 = *'1999s131'*, x_4 = *'alice'*, x_5 = *'rossi'*.
2. *Collaborator(ABC, Alice,* x_1, x_2, x_3, x_4*)* :- x_1 = *'stage'*, x_2 = *'IS137'*, x_3 = *'alice' ,* x_4 = *'rossi'*.

3. *University(MIUR, XYZ,* x_1*) :-* x_1 *= 'University of XYZ'.*
4. *University(CCE, Y,* x_1*) :- University(MIUR, Y,* x_1*).*
5. *Student(CCE, Y,* x_1, x_2, x_3, x_4, x_5*) :-*
   University(CCE, X, y_1*),* $y_1 = uni_{name}$*, Student(X, Y,* x_1, x_2, x_3, x_4, x_5*),* $x_1 = uni_{name}$*.*
6. *ABCGuest(CCE, Y) :-*
   Collaborator(ABC, Y, x_1, x_2, x_3, x_4*),* $x_3 = name_{first}$*,* $x_4 = name_{last}$*,*
   Student(CCE,Y, y_1, y_2, y_3, y_4, y_5*),* $y_4 = name_{first}$*,* $y_5 = name_{last}$*.*
7. *Guest(CCE, Y) :-*
   Student(CCE,Y, y_1, y_2, y_3, y_4, y_5*).*

The RTML PDP reads from a configuration file that, for the service that Alice requested, the role *ABCGuest* or *Guest* is required. Then, the RTML PDP invokes the Prolog Engine with the previous statements to perform the evaluation of the goals:

1. $?:-ABCGuest(CCE, Alice)$.
2. $?:-Guest(CCE, Alice)$.

Alice has the access to *ManagedJobFactoryService* granted by his credentials by both the roles and she can choose the one to use in the JobConfigurationFile (the default is the role with the lowest AccessLevel value). If the second credential is not owned by Alice because she is not a ABC collaborator, the second goal is not verified, and she can only access the service with the *Guest* role. This role will have some limitation with respect to the privilege granted to a ABC collaborator.

5 Related Work

This section describes other attempts to enhance the Globus Security Infrastructure by integrating advanced authorization systems.

The Community Authorization Service, CAS, has been proposed by the Globus team in [3] and [10] to improve the standard authorization service. CAS is a service that stores a database of VO policies, i.e. the policies that determine what each Grid user is allowed to do as VO member. This service issues to Grid users proxy certificates that embed CAS policy assertions. These credentials will be presented by the Grid user to the service it wants to exploit. From the point of view of the Grid resource provider, the gridmap file includes one entry only, that pairs the DN of the CAS with a local user. Hence, this solution uses the same local user to log all the Grid users that belong to the same CAS, and globus must be able to understand and enforce the policies included in the credentials released by the CAS. However, these policies are coarse grained, because they only decide which of the local services can be accessed by the grid user. Moreover, in this solution a Grid user can only enjoy trust granted from their membership to a specific CAS, which is independent from his previous behavior in accessing other Grid sites.

An alternative approach that integrates an existing authorization system in the Grid environment has been proposed by Keahey and Welch. In [4], they describe some of the shortcomings of the current Globus authorization system

and they state the need for a more powerful authorization system. In [13], they address this issue by integrating Akenti within the Globus toolkit. "Akenti is an authorization service (PDP) that uses authenticated X.509 certificates to establish identity and distributed digitally signed authorization policy certificates to make access decisions about distributed resources" [14]. Each certificate includes the attributes assigned by the VO to the Grid user. The JobManager finds out from the resource policy the attributes required to access the resource, and matches them with the ones owned by the Grid user.

Another solution to adopt an advanced authorization system in the Grid environment has been presented by Stell, Sinnot and Watt in [12]. They integrate a role based access control infrastructure, PERMIS, with the Globus toolkit to provide control on user rights. PERMIS is "a role based access control infrastructure that uses X.509 certificates to store users' role. All access control decisions are driven by an authorization policy, which is itself stored in a X.509 attribute certificate.." [1]. PERMIS supports classical hierarchical RBAC, in which roles are allocated to users and privileges to roles. Its limit depends on the weakness of role's relationships in which senior roles inherit privileges from junior roles. These relations are not always satisfactory to express complex policies.

The previous two solutions provide an attribute-based and a role-based access control system, but they do not provide a proper mechanism to map a Grid user onto a local user whose set of rights is determined according to the attributes or to the role.

6 Conclusions and Future Work

This paper proposed to integrate in Globus a flexible and powerful authorization system that enables each site to dynamically evaluate user's requests exploiting his credentials. At this time we experimented the RTML only for user authorization and once the requested job is in execution, any other restriction is provided than the one of the Unix account system. A future goal could extend our framework in a way such that a role also determines a security policy that is enforced during the execution of the application.

Moreover the assumption that all credentials are stored in a fixed set of public repositories could not always be satisfactory for those applications which require the use of trust management for decentralized control, like the Grid environment. A future development will be the implementation of a distributed credential chain discovery that will consider how to collect all the credentials. The evaluation can start with an incomplete set of credentials, then can be suspended waiting for a new set of credentials, and then resumed when additional credentials are obtained. This step could be iterated as needed.

Acknowledgments. We would like to thank Ninghui Li for providing us a beta version of the parser for RTML credentials we used in our implementation.

References

1. Chadwick, D.W., Otenko, A.: The permis x.509 role based privilege management infrastructure. In: Proceedings of the seventh ACM symposium on Access control models and technologies (SACMAT 2002), pp. 135–140. ACM Press, New York (2002)
2. Foster, I.: Globus toolkit version 4: Software for service-oriented systems. In: Jin, H., Reed, D., Jiang, W. (eds.) NPC 2005. LNCS, vol. 3779, pp. 2–13. Springer, Heidelberg (2005)
3. Foster, I., Kesselman, C., Pearlman, L., Tuecke, S., Welch, V.: A community authorization service for group collaboration. In: Proceedings of the 3rd IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'02), pp. 50–59. IEEE Computer Society Press, Los Alamitos (2002)
4. Keahey, K., Welch, V.: Fine-grain authorization for resource management in the grid environment. In: Parashar, M. (ed.) GRID 2002. LNCS, vol. 2536, pp. 199–206. Springer, Heidelberg (2002)
5. Li, N., Mitchell, J., Winsborough, W.H.: Beyond proof-of-compliance: Safety and availability analysis in trust management. In: IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Alamitos (2003)
6. Li, N., Mitchell, J.C.: Datalog with constraints: A foundation for trust management languages. In: Dahl, V., Wadler, P. (eds.) PADL 2003. LNCS, vol. 2562, pp. 58–73. Springer, Heidelberg (2002)
7. Li, N., Mitchell, J.C., Winsborough, W.H.: Design of a role-based trust management framework. In: Proceedings of IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitos (2002)
8. Li, N., Tripunitara, M.V.: Security analysis in role-based access control. In: Proceedings of the ninth ACM Symposium on Access Control Models and Techniques (SACMAT 2004), ACM Press, New York (2004)
9. Li, N., Winsborough, W.H., Mitchell, J.C.: Distributed credential chain discovery in trust management. Journal of Computer Security 1, 35–86 (2003)
10. Pearlman, L., Kesselman, C., Welch, V., Foster, I., Tuecke, S.: The community authorization service: Status and future. In: Proceedings of Computing in High Energy and Nuclear Physics (CHEP03): ECONF, C0303241:TUBT003 (2003)
11. Globus authorization framework: Saml callout, `http://www.globus.org/toolkit/docs/4.0/security/authzframe/`
12. Stell, A.J., Sinnott, R.O., Watt, J.P.: Comparison of advanced authorisation infrastructures for grid computing. In: Proceedings of High Performance Computing System and Applications 2005, HPCS, pp. 195–201 (2005)
13. Thompson, M.R., Essiari, A., Keahey, K., Welch, V., Lang, S., Liu, B.: Fine-grained authorization for job and resource management using akenti and the globus toolkit. In: Proceedings of Computing in High Energy and Nuclear Physics (CHEP03) (2003)
14. Thompson, M., Essiari, A., Mudumbai, S.: Certificate-based authorization policy in a pki environment. ACM Transactions on Information and System Security (TISSEC) 6(4), 566–588 (2003)
15. Winsborough, W.H., Li, N.: Safety in automated trust negotiation. In: IEEE Symposium on Security and Privacy, IEEE Computer Society Press, Los Alamitos (2004)

GRID Software Solution for the Segmentation of the Coronary Artery Tree in Biplane Angiograms

Bernhard Quatember and Martin Mayr

Innsbruck Medical University, Clinical Department of Radiology II,
Anichstrasse 35, 6020 Innsbruck, Austria
{Bernhard.Quatember,Martin.Mayr}@uibk.ac.at

Abstract. We are developing a computer system that will enable the precise diagnosis and the planning of therapy in the field of coronary artery disease. The implementation of this system is carried out within the framework of a computational GRID (Austrian GRID). Our aim is to obtain patient-specific simulations of coronary hemodynamics. The geometry of the flow domain is derived from biplane angiograms. In these images, the coronary artery tree must be segmented. Thereafter, a three-dimensional reconstruction method is applied. However, biplane angiograms are subject to noise, blur, and image deterioration caused by background structures. In order to eliminate these disturbances, several image pre-processing steps and a newly developed segmentation method must be carried out. In this paper we will concentrate on our new segmentation approach and show that the segmentation of the coronary artery tree can be solved advantageously within the framework of a computational GRID.

Keyword: Computational GRID, coronary hemodynamics, simulation, medical imaging, parallel computing.

1 Introduction

In the field of medicine, computational GRIDs are not yet widely in use. This is particularly true for applications to clinical practice. Here we will deal with the development of a GRID-based computer system for the domain of interventional cardiology, a field in which the use of GRID technology is completely new [1]. For the treatment of coronary artery disease, interventional cardiologists and heart surgeons already have excellent technical facilities at their disposal. The devices needed for bypass operations, balloon angioplasty and the placement of stents are already highly refined and it is unlikely that fundamental improvements will be made in them in the foreseeable future. In contrast, contemporary methods for making diagnoses, clinical decisions, and plans for treatment are much less advanced. As a consequence of these inadequacies, it seems likely that the most effective and efficient way to improve the quality of care would be the development of a computer system that enables cardiologists, among other things, to

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 457–464, 2007.
© Springer-Verlag Berlin Heidelberg 2007

– quantitatively assess the patient's hemodynamics and
– quantitatively predict the changes in (improvements of) the perfusion of the myocardium that can be attained by therapy.

It is the central aim of our ongoing research to develop a computer system which will allow cardiologists

– to simulate the patient-specific coronary hemodynamics before therapy and to simulate the manner in which they would be altered by the therapy that is being taken into consideration, and
– to carry out parameter identification tasks and determine clinically relevant parameters.

We discovered that the tasks which would have to be carried out by this computer system are computationally extremely expensive. Moreover, cardiologists in clinical settings cannot wait very long for the results. In order to tackle these computational challenges, we envisaged a computational GRID solution from the outset. In particular, we decided to implement the software within the framework of the "AUSTRIAN GRID", a computational GRID established and supported by the Austrian Federal Ministry for Science and Research. The main advantages of a computational GRID in our problem area are:

– that it accords all the physicians (cardiologists) within a geographical region full and equal access to the system's facilities and
– its potentiality to enable the parallel processing for our computationally extremely expensive tasks.

We will first give an overview of the computer system and then concentrate on the acquisition of the geometry of the patient's coronary (epicardial) arteries, which is derived from biplane angiograms. The acquisition of the geometry is a prerequisite for our simulation studies of the coronary hemodynamics. We will show that these acquisition procedures involve several complicated and time-consuming steps. Hence, the parallelism offered by our computational GRID is urgently needed to solve this task.

2 Overview of Our Computer System and General Remarks on the Development of Our Software

Our simulation efforts comprise

– three-dimensional simulation studies of the coronary blood flow (based on the finite element method) and on
– simulations involving lumped parameter models.

Our simulation studies are patient-specific; they fully take the geometry of the flow domain within the patient's coronary arteries into consideration.

The geometry of the flow domain is usually derived from biplane angiograms of the patient. These are taken in catheterization laboratories; they are high-resolution

images of the coronary (epicardial) arteries after they have been filled by a contrast medium. Although biplane angiography is an invasive examination, it nevertheless serves as the "gold standard" among all imaging modalities that are applicable in the field of cardiovascular imaging. To derive the geometry of the flow domain, complicated image processing techniques must be applied, and moreover it is necessary to accomplish a three-dimensional reconstruction [2]. The software for the system we are developing thus includes both software modules for the simulation of the coronary blood flow and ones for the processing of medical images.

2.1 Software Modules for the Simulation of the Coronary Blood Flow

Our aim is to conduct patient-specific simulation studies based on the aforementioned medical imagery. In particular, we are carrying out three-dimensional computer simulation studies of the flow of blood within the epicardial arteries, especially around stenoses (narrowed arterial segments), since the knowledge of the flow pattern in such regions of disturbed flow is highly important for the assessment of the severity of the coronary artery disease. We use the commercial CFD software package FIDAP which is based on the finite element method. However, due to the high complexity of the network of the coronary vessels, it is almost impossible to simulate the flow within the entire coronary system in this manner. Our three-dimensional simulation studies are thus more or less confined to regions with disturbed flow. To obtain a global assessment of the coronary hemodynamics, we preferably use lumped parameter simulation models. Models of this kind lead to systems of (nonlinear) ordinary differential equations and therefore require less computational effort than three-dimensional simulations. We provide for a relatively large number of lumped components. Our approach proved to be sufficiently accurate to quantitatively simulate the overall flow in the coronary arteries and the supply of blood to the myocardium. In any case, however, a prerequisite for our patient-specific simulations is a fair knowledge of the geometry of the flow domain. As mentioned earlier, this knowledge can be derived from biplane angiograms by using software modules which will be subsequently described.

2.2 Software Modules for Image Processing and Three-Dimensional Reconstruction

In current clinical practice, cardiologists do not have a representation of the genuine three-dimensional geometry of the epicardial arteries at their disposal, and are therefore not able to quantitatively assess the severity of stenoses. Moreover, they frequently overlook diffusely narrowed segments. We have developed software modules that allow us to calculate a three-dimensional representation of the tree-like structure of the coronary (epicardial) arteries. Furthermore, our approach is suitable for use in clinical settings, since it facilitates the largely automatic handling of the angiograms; in particular, it helps to reduce the amount of human-computer interaction to a minimum. An important characteristic of our approach is the largely automatic segmentation of the coronary arteries in biplane angiograms. This is rather difficult to achieve, since we have to reckon

with noise, artifacts, blur and other flaws which deteriorate the image quality. To confront the challenge, we developed an advanced segmentation technique using a priori knowledge, which we will describe in the following section.

3 Segmentation of the Angiograms

As mentioned earlier, complicated image processing methods must be applied to segment the tree-like structure of the coronary (epicardial) arteries in biplane angiograms. To be of clinical value, the segmentation procedures must be carried out largely automatically. As contemporary image processing products proved to be insufficient for our requirements, we developed a segmentation method requiring a number of procedural steps, the most important of which will be described below.

3.1 Hessian Artery Enhancement Filtering

Our segmentation technique makes use of the a priori knowledge that the coronary (epicardial) arteries have a tree-like tubular structure. It applies differential geometric criteria to the identification of the tubular structures of the contrast-medium-filled coronary (epicardial) arteries and distinguishes them from other structures [3]. For this purpose, we developed a Hessian artery enhancement filter which operates by using a multi-scale filtering approach. The filtering process consists of two steps:

- In the first step, the angiograms are convolved with multi-scale Gauss filters. The Gaussian smoothing is used to suppress noise; the degree of smoothing is determined by applying different values of the standard deviation σ of the Gaussian filter:
$$\Sigma = \{\sigma_1, \sigma_2, .., \sigma_n\} \tag{1}$$
The typical range of n is
$$3 \leq n \leq 8 \tag{2}$$
- In the second step, the eigenvalues and eigenvectors of the Hessian matrix are calculated for each smoothed image. These calculated results describe the second-order variations of the image intensity and are well-suited for the detection of tubular structures.

With the aforementioned eigenvalues of the Hessian matrix, a so-called vesselness function is calculated as follows:

For a given value of $\sigma \in \Sigma$ (single scale filtering), the Hessian matrix $H(\boldsymbol{x})$ and their eigenvalues $\lambda_1(\boldsymbol{x})$ and $\lambda_2(\boldsymbol{x})$ are functions of the position $\boldsymbol{x}$ of the image. In the case of tubular structures, the following relation for the eigenvalues holds:

$$|\lambda_1| \gg |\lambda_2| \tag{3}$$

In our approach, we use the so-called vessel function as defined in [3]

$$V(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \lambda_1 < 0, \\ exp(-\frac{{R_B}^2}{2{\beta_1}^2})[1 - exp(-\frac{S^2}{2{\beta_2}^2})] & \text{otherwise} \end{cases} \tag{4}$$

in which

$$R_B = \frac{|\lambda_2|}{|\lambda_1|} \tag{5}$$

and

$$S = \sqrt{\lambda_1^2 + \lambda_2^2} \tag{6}$$

As mentioned above, we have chosen a multi-scale approach for our filtering with varying standard deviations σ. Hence, we calculate a family of vesselness functions $V_\sigma(\boldsymbol{x})$ with σ as a parameter. We then compute a maximum intensity projection of this family $V_\sigma(\boldsymbol{x})$. Based on it, we generate a binary mask (thresholding) and then apply a thinning filter. As a result, we obtain a skeletonized image.

3.2 Detection of Border Lines

The second phase of our segmentation process involves the application of algorithms to detect the border lines (edges) of the representation of the coronary arteries in our angiograms [4]. The aforementioned skeleton image serves as the basis for this detection. First we carry out a smooth approximation to the discrete set of points (pixels) of the skeleton image by calculating splines, which are regarded as preliminary centerlines of the coronary arteries. The tree-like structure of these splines is then superimposed onto the original image (angiogram). At relatively short intervals we draw normals to these splines, which we call scan lines. Along each scan line of this family of lines we acquire and analyze the intensity profile. In this analysis, we confine ourselves to an interval of the scan line that is only marginally larger than the maximally expected size of the coronary (epicardial) artery. Mainly due to the finite focal spot size (apparent focal spot size) of the X-ray tubes, the relatively small coronary arteries will be considerably distorted by geometric blurring. In order to achieve a sophisticated analysis of these intensity profiles and, especially, the deblurring of the angiograms, we have developed model-based border detection algorithms and, moreover, specific software components to suppress noise and background structures. These software routines are highly demanding in terms of CPU time. As already mentioned above, the main difficulty in detecting the border lines of the coronary (epicardial) arteries in biplane angiography is the lack of sharpness (blur). To avoid a considerable reduction of radiographic clarity as a result of motion, X-ray exposures of high intensity and short duration (about 5 ms) are used. As a consequence, the focal spot size of the grid-controlled tubes for the generation of such pulsed x-rays cannot be made arbitrarily small: typical focal spot sizes (apparent focal spot sizes) are in the range of about 0.4 to 0.8 mm. However, a focal spot size in this range causes a considerable geometric

blur. In our analysis of the intensity profile we must thus consider the focal spot size and, in addition, the intensity distribution of the X-rays along the focal spot in transversal direction, which is usually formulated mathematically as the line-spread function $L(x)$. In our newly-developed border detection method the following model-based approach [4,5] is applied: We compare a particular acquired intensity profile with a family of intensity profiles calculated for arteries with diameters that vary within a pre-specified diameter range. For each calculated intensity profile, we determine the sum of the squares of the differences between the intensity values of the acquired profile in the original image and the corresponding intensity values of the calculated profiles. Among the individual profiles of the family we then select the profile with the minimal sum value. The diameter assigned to this profile is regarded as the blur-corrected diameter of the artery. For details, refer to [4].

4 Exploitation of the Parallelism Offered by the GRID Environment

The above-described image-processing procedures and all the other algorithms of our system are computationally extremely expensive, even for a single angiogram [2]. A condensed activity diagram of the entire work flow of these image processing activities can be seen in Figure 1. In each angiogram the left and the right coronary arteries and their branches can be processed independently (cf. Figure 1).

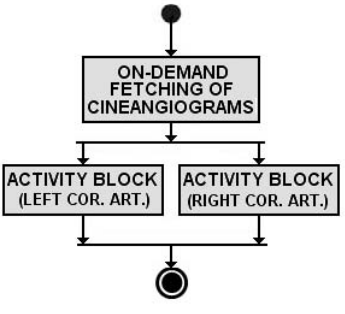


Fig. 1. Condensed activity diagram of the entire workflow

In Figure 1, "Activity Block (LEFT COR. ART.)" comprises all the image processing and all of the supplementary activities that are relevant for the left coronary artery and its branches, "Activity Block (RIGHT COR. ART.)" is an identical replicate of "Activity Block (LEFT COR. ART.)"; it includes all the activities that must be applied to the right coronary artery and its branches.

Moreover, in biplane angiography, we have at least two angiograms (projections A and B) which can also be processed independently. Furthermore, we

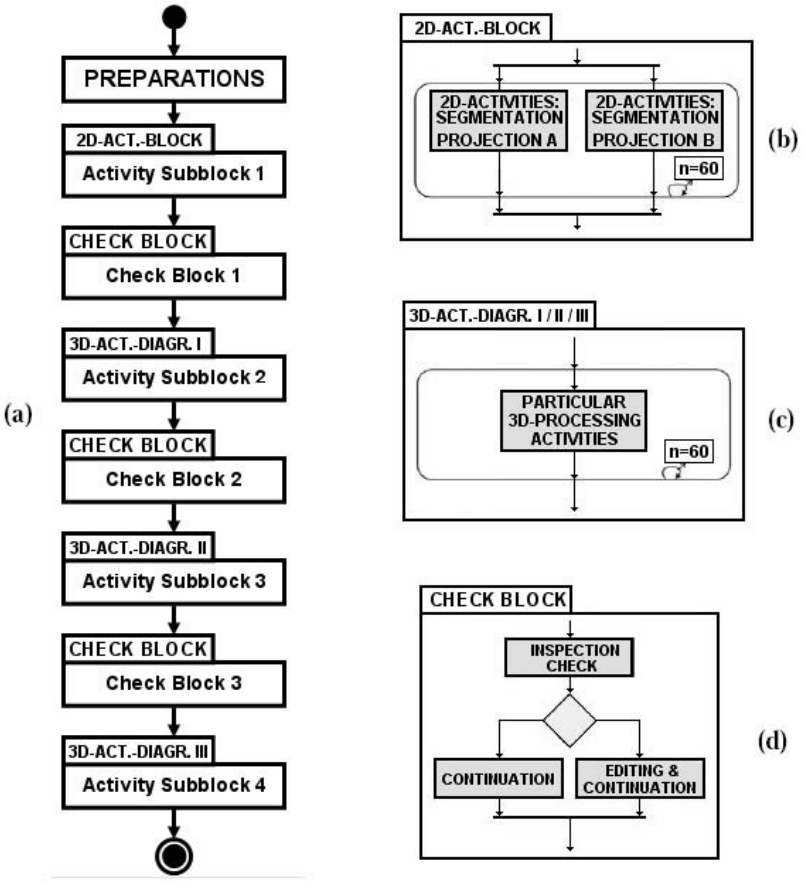


Fig. 2. "Activity Subblocks" and "Check Blocks"

have to bear in mind that usually cineangiograms are taken. A cineangiographic study usually consists of about 60 pairs of angiograms, all of which must be processed; the image processing can thus be subdivided into about 120 independent tasks (one for each angiogram) that can be carried out in parallel [6]. This coarse-grain parallelism which involves no interdependence between the individual tasks is well suited for utilization of the clusters of workstations incorporated in the Austrian GRID. Further blocks depicted in Figure 1 are the fetching of the cineangiograms, the construction of 3D geometric models, the generation of meshes, and the motion analyses of the left and right coronary artery trees.

Each of the two "Activity Blocks" of Figure 1 comprehends the "Activity Subblocks" 1 to 4 and the "Check Blocks" 1 to 3. The activities in "Activity Subblock" 1 comprise the above described segmentation processing steps (projections A and B). "Activity Subblock" 2 handles the task of three-dimensional reconstruction. "Activity Subblock" 3 includes all the activities needed for the

creation of a wireframe model. "Activity Subblock" 4 comprises the activities for the mesh generation [2] required by the finite element method. Each of the "Activity Subblocks" 1 to 3 forwards its results to a "Check Block" ("Check Blocks" 1 to 3), which allows the cardiologist to check and, if necessary, to edit (correct) the results (e.g. in case of imperfect medical imagery). The sequence of the activities within an entire "Activity Block" (cf. Figure 1) is depicted in the block diagram of Figure 2a. Figure 2b shows a condensed activity diagram relevant to "Activity Subblock" 1, whereas Figure 2c is relevant to "Activity Subblocks" 2 to 4. The activity diagram of Figure 2d is relevant to all of the "Check Blocks."

Acknowledgements

The work described in this paper is partially supported by the "Austrian GRID" project, funded by the Austrian BMWF (Federal Ministry for Science and Research) under contract GZ 4003/2-VI/4c/2004./

References

1. Quatember, B., Mayr, M., Muehlthaler, H.: Clinical usefulness of a computational grid for diagnosis and planning therapy of coronary artery disease. In: Volkert, J., Fahringer, T., Kranzlmueller, D., Schreiner, W. (eds.) Proceedings of the 1. Austrian Grid Symposium, Schloss Hagenberg, December 01 - 02, 2005. books@ocg.at, Oesterreichische Computer Gesellschaft, vol. 210, pp. 75–89 (2006)
2. Quatember, B., Muhlthaler, H.: Generation of cfd meshes from biplane angiograms: an example of image-based mesh generation and simulation. Applied Numerical Mathematics 46(3-4), 379–397 (2003)
3. Schrijver, M., Slump, C.: Automatic segmentation of the coronary artery tree in angiographic projections. In: Proceedings of BroRISC 2002, Veldhoven, cNetherlands, pp. 449–464 (November 28-29, 2002)
4. Mayr, M., Quatember, B.: Development of a special method and a software system for the semi-automatic segmentation of biplane angiograms. In: Volkert, J., Fahringer, T., Kranzlmueller, D., Schreiner, W. (eds.) Proceedings of the 2. Austrian Grid Symposium, Innsbruck, Austria, September 21-23, 2006. books@ocg.at., Oesterreichische Computer Gesellschaft, vol. 221, pp. 220–237 (2007)
5. Chan, R.C., Karl, W.C., Lees, R.S.: A new model-based technique for enhanced small-vessel measurements in x-ray cine-angiograms. IEEE Trans Med Imaging 19(3), 243–255 (2000)
6. Fahringer, T., Prodan, R., Trawoeger, B., Quatember, B., Mayr, M.: Workflow modelling of grid-based system for diagnosis of coronary artery disease with agwl. In: Volkert, J., Fahringer, T., Kranzlmueller, D., Schreiner, W. (eds.) 1st Austrian Grid Symposium. Volume 210., Schloss Hagenberg, Austria, Oesterreichische Computer Gesellschaft (books@ocg.at), vol. 210, pp. 234–244 (2006)

An Intelligent Scatter with Estimation of Distribution for Tabu Search

Masaharu Munetomo[1], Yuta Satake[2], and Kiyoshi Akama[1]

[1] Information Initiative Center
{munetomo,akama}@iic.hokudai.ac.jp
[2] Graduate School of Information Science and Technology,
Hokkaido University, Sapporo 060-0811, Japan
satake@uva.cims.hokudai.ac.jp

Abstract. This paper proposes an efficient optimization algorithm based on tabu search and estimation of distribution algorithms. The proposed algorithm estimates characteristics of distribution of solutions after performing tabu searches based on a marginal product model to acquire linkage information. Once correct linkage information is obtained, we can perform crossovers effectively without disrupting building blocks using the information. The proposed algorithm is expected to adapt to both global and local characteristics of solution space. Through empirical studies, we show the effectiveness of our approach.

Keywords: tabu search, extended compact genetic algorithm, estimation of distribution algorithms.

1 Introduction

In this paper, we propose an efficient optimization algorithm employing tabu search[1] and advanced evolutionary algorithms called estimation of distribution algorithms (EDAs)[4]. We intend to replace scatter search in multi-start tabu search by the learning with an EDA; the purpose of the learning is to generate preferable starting points that guide tabu search to global optima. EDAs are proposed to improve robustness of evolutionary algorithms such as genetic algorithms (GAs). EDAs build their probabilistic models based on current promising solutions, which is employed to generate offspring. In this paper, we deal with an EDA technique called extended compact genetic algorithm (ECGA) to be combined with tabu search. We show the effectiveness of our approach through numerical experiments on some typical test functions.

2 Extended Compact Genetic Algorithm

Extended compact genetic algorithm (ECGA) is proposed by Harik, et. al.[2]. The ECGA is based on the minimum description length (MDL) principle to identify linkage sets of loci from distribution of alleles in a population of binary strings. The probabilistic model employed in the ECGA is marginal distribution model (MPM) determined by the linkage sets.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 465–472, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The ECGA performs the following algorithm[2].

1. Create a random population of N binary strings of length L.
2. Apply selection.
3. Model the population using a greedy MPM search.
4. Generate a new population according to the MPM found in step 3.
5. If stopping criteria is not satisfied, return to step 2.

Its model building process starts from sets of independent locus, that is, [1][2]...[L] and a greedy search is performed to minimize the following complexity measure $C = C_m + C_p$ based on the MDL principle:

$$C_m = \log N \sum_i 2^{S[i]}, \tag{1}$$

$$C_p = N \sum_i E(M[i]), \tag{2}$$

where N is the number of population, $S[i]$ is the length of the i-th linkage set, $M[i]$ is the probabilistic distribution of marginal probabilities based on the i-th linkage set, and $E(M[i])$ is the entropy of the probabilistic distribution $M[i]$.

The greedy MPM search maximizes the complexity C according to the following sequence to obtain linkage sets:

1. Initialize linkage sets consisting of independent locus.
2. For all the pairs of linkage sets, calculate their ΔC, differences of C by merging them.
3. If $\Delta C \geq 0$ for all the pairs, terminate the search.
4. Otherwise, select a pair of linkage sets that minimizes ΔC and merge them to generate a larger linkage set.
5. Goto 2.

ECGA performs crossovers based on the obtained linkage sets. Simple crossovers should disrupt useful sub-components of solutions called building blocks, when we cannot encode them tightly onto a string. By the learning through greedy MPM search based on the complexity measure, we can avoid to disrupt building blocks, which leads to obtain optimal solutions quite effectively when tight encoding of building blocks is not ensured.

3 Multi-start Tabu Search with Estimation of Distribution

We propose an efficient method for multi-start tabu search replacing scatter search by estimation of distribution. Scatter search is a technique to produce initial points for multi-start local search by generating linear combination of a pair of current local optima. This approach is effective when linear combination

of current local optima leads to find global ones. Otherwise, we should seek for another, more advanced approach to set initial points of local searches.

We introduce a machine learning approach to find *promising* initial points for tabu search by introducing EDAs. Instead of just selecting promising solutions from current population of strings (solutions), we generate probabilistic models from a set of local optima found by multi-start tabu search. We build probabilistic models from the set of local optima to generate offspring to be initial points to restart tabu searches.

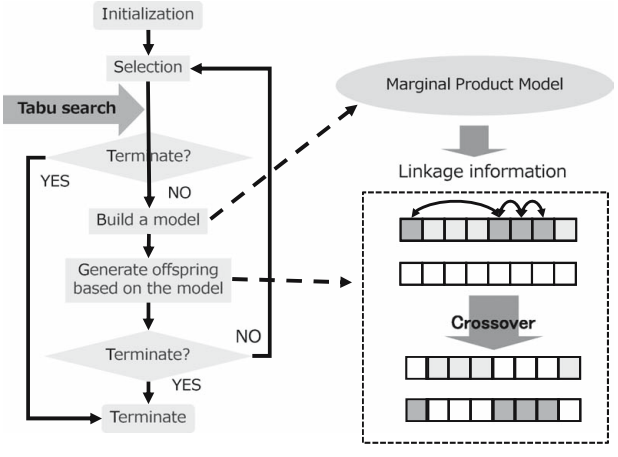


Fig. 1. Overview of the ECGA-Tabu hybrid

Figure 1 shows an overview of the proposed algorithm. The algorithm performs tabu search for each solution before starting model-building process of the ECGA which is applied to all the strings in a population to obtain linkage sets and generate offspring by performing crossovers based on the linkage information. Based on the model that minimizes complexity measures, we can avoid disruption of building blocks even when building blocks are loosely encoded on a string.

The following is the detailed algorithm of the ECGA-Tabu hybrid we propose:

1. Generate N initial solutions $\{s_1, \cdots, s_N\}$ randomly.
2. Select M $(< N)$ solutions from the current solutions.
3. Perform the following for $p = 1, \cdots, M$.
   (a) Tabu search is performed by starting from each solution s_p.
   (b) When it is converged, replace s_p by it.
4. If a terminate solution is satisfied, finish. Otherwise, continue.
5. Perform ECGA on the current solutions $\{s_1, \cdots, s_M\}$:
   (a) generate a MPM model from $\{s_1, \cdots, s_M\}$.
   (b) generate offspring $\{s_1, \cdots, s_N\}$ by performing crossovers based on the MPM model.
6. If a terminate condition is satisfied, finish. Otherwise, goto 2.

Note that we check terminate conditions after both tabu search and ECGA. This is because an optimal solution may be obtained by both of them.

4 Empirical Results

We perform numerical experiments to compare the following algorithms:

- Hill-climbing (HC)
- Tabu search (TS)
- Extended Compact GA (ECGA)
- ECGA with Hill-climbing (ECGA+HC)
- ECGA with Tabu search (ECGA+TS)

Hill-climbing is a simple local search without any mechanism of escaping from local optima, and therefore, it cannot solve multimodal problems effectively. Tabu search can escape from some of local optima by employing tabu list. We have prosed a hybrid algorithm consisting of ECGA and simple hill-climbing elsewhere[3]. Resulting algorithm is called ECGA with Hill-climbing (ECGA+HC). They are compared with ECGA with tabu search we propose in this paper.

We employ the following three test functions for the experiments.

- The sum of 5-bit trap functions
- The sum of 6-bit w-trap functions
- Rastrigin function

The sum of 5-bit trap functions, which is frequently employed to test GAs, is defined as follows.

$$F_{trap5}(s) = \sum_{i=1}^{n} trap_5(u_i), \tag{3}$$

$$trap_5(u) = \begin{cases} 4-u & \text{if } u = 0 \sim 4 \\ 5 & \text{if } u = 5, \end{cases} \tag{4}$$

where u_i is unitation (the number of 1) of the i-th 5-bit substring s_i ($s = s_1 s_2 \cdots s_n$).

Figure 2 illustrates the 5-bit trap subfunction. Each subfunction has one global optimum and one local optimum. A deceptive attractor from $u = 4$ to $u = 0$ misleads the search to a local optimum. The overall sum of the subfunction has one global optimum and $2^n - 1$ local optima. This means that the test function is highly multimodal where the number of local optima grows in an exponential order of string length.

The sum of 6-bit w-trap function, which is designed to add more complexities in fitness landscape of the 5-bit trap test function, is defined as follows:

$$F_{wtrap6}(s) = \sum_{i=1}^{n} wtrap_6(u_i), \tag{5}$$

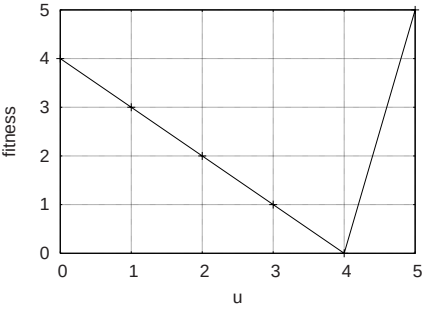


Fig. 2. 5-bit Trap function

$$wtrap_6(u) = \begin{cases} 5 & \text{if } u = 0, \\ 2u - 2 & \text{if } u = 1 \sim 3 \\ 10 - 2u & \text{if } u = 3 \sim 5 \\ 6 & \text{if } u = 6 \end{cases} \tag{6}$$

where u_i is unitation (the number of 1) of the i-th 6-bit substring s_i ($s = s_1 s_2 \cdots s_n$).

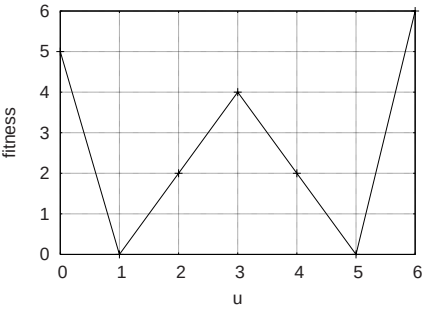


Fig. 3. 6-bit W-Trap function

Figure 3 illustrates the 6-bit w-trap subfunction. Each subfunction has one global optima at $u = 6$ and 21 local optima at $u = 0$ and $u = 3$ since we have one local optimum at $u = 0$ and ${}_6C_3$ local optima at $u = 3$. Deceptive attractors from $u = 1$ to $u = 3$ and $u = 5$ to $u = 3$ misleads the search to a local optimum. The overall sum of the subfunction has one global optimum and $21^n - 1$ local optima. This means that the test function is extremely highly multimodal.

Rastrigin function is defined as follows:

$$\text{Rastrigin: } f(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10],$$
$$x_i \in [-5.12, 5.12] \text{ (10bit)}$$

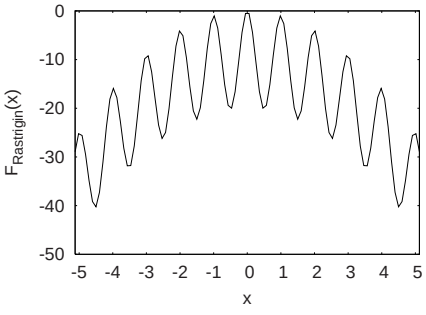


Fig. 4. Rastrigin function

Figure 4 shows a landscape near the optimum of Rastrigin function.

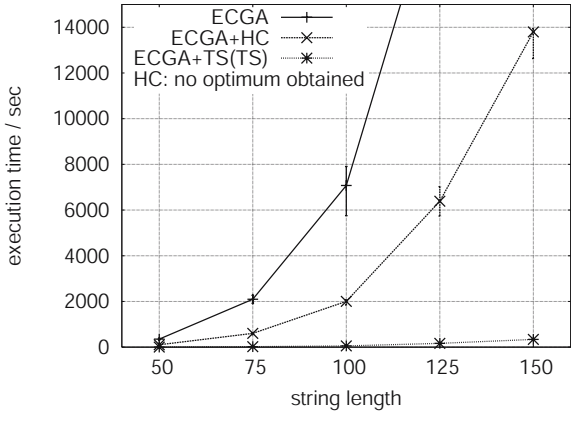


Fig. 5. Comparisons of execution times for the trap function

Figure 5 shows a comparison of execution times for the trap test function. The x-axis shows the length of the strings and the y-axis shows execution times to obtain optimal solutions. The result shows that simple hill-climbing cannot solve the problem which is a highly multi-modal function. Other algorithms can obtain optimal values for this test function and ECGA+Tabu and Tabu search achieve the best results. This is because in each trap subfunction, once the search reaches to a local optimum ($u = 0$), tabu list guides the search reversely from $u = 0$ to $u = 4$ along the deceptive attractor of the subfunction. This is simply caused by the characteristics of the subfunction which has only one local optimum and one global optimum.

This is the reason we design and employ a w-trap function by increasing the number of local optima in each trap subfunction. $2k$-bit w-trap function has local optima when $u = k$ in addition to when $u = 0$. In this function, we have $1 +_{2k} C_k$ local optima in each subfunction.

Figure 6 shows the comparison of execution times for the w-trap test function.

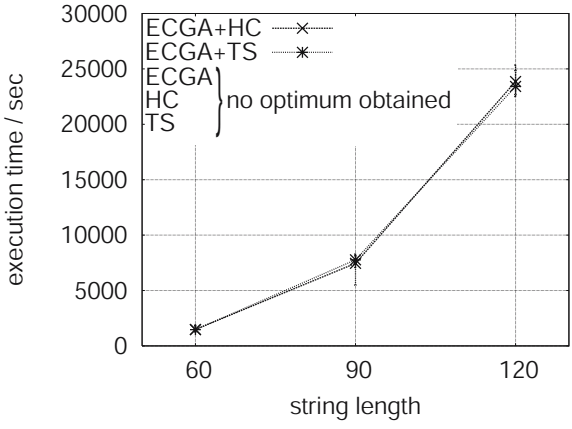


Fig. 6. Comparisons of execution times for the w-trap function

This result indicates that tabu search cannot obtain optimal solutions for this test function and ECGA alone cannot obtain optimal solutions, either. In order to obtain optimal solutions, we need a hybrid algorithm of ECGA and local search. Both ECGA+HC and ECGA+Tabu can obtain optimal solutions with almost the same computational cost.

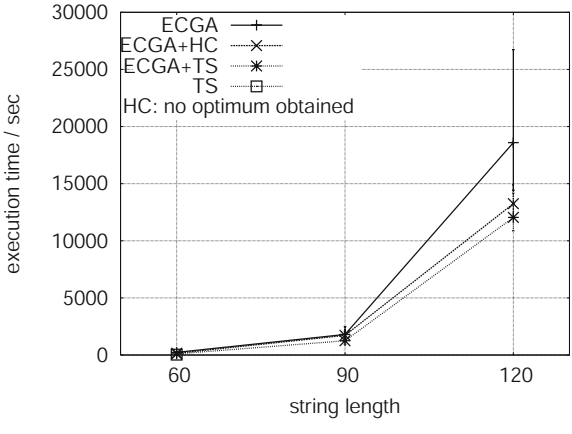


Fig. 7. Comparisons of execution times for the Rastrigin function

Figure 7 shows a comparison of computation times for Rastrigin function. For this function, simple hill-climber cannot obtain optimal solutions due to the multimodality of the function. Tabu search can obtain optimal solutions only when string length is 60. Increasing string length, Rastrigin function becomes difficult to optimize for local search techniques. ECGA and its local search hybrids can

obtain optimal solutions for all the cases, and ECGA+Tabu achieves the best result. This is because ECGA adapts to complex landscape of the objective function and tabu search searches locally escaping from some local optima toward a global optimum.

5 Discussion and Conclusion

Evolutionary algorithms such as GAs are usually employed together with local search techniques. GAs are considered global search techniques that adapt to global structure of solution spaces based on building blocks. A combination of global and local searches should be effective for real-world application problems which usually have global and local landscapes in their solution spaces.

In this paper, we propose a local and a global search hybrid consisting of tabu search and ECGA, both of which are advanced methods in local and global search techniques. The resulting ECGA+Tabu algorithm is a promising approach for difficult optimization problems consisting of global structure that adapts to advanced methods in evolutionary algorithms employing machine learning technique, and local structure that adapts to tabu search. The proposed algorithm is a promising approach to solve wide-spectrum of real-world application problems.

Through experimental studies, we show the effectiveness of our approach compared with tabu search and ECGA alone. We can obtain optimal solutions quite effectively for the entire test functions, while others cannot obtain optimal solutions or can obtain them with more computational overheads.

References

1. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 5, 533–549 (1986)
2. Harik, G.: Linkage Learning via Probabilistic Modeling in the ECGA. IlliGAL Report No. 99010, University of Illinois at Urbana-Champaign, Urbana, IL (1999)
3. Munetomo, M., Satake, Y., Akama, K.: Enhancing model-building efficiency in extended compact genetic algorithms. Proceedings of the 2006 IEEE Conference on Systems, Man, and Cybernetics 10, 2362–2367 (2006)
4. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithms. Kluwer Academic Publishers, Dordrecht (2002)

A Framework of GRID Problem-Solving Environment Employing Robust Evolutionary Search

Masaharu Munetomo[1], Asim Munawar[2], and Kiyoshi Akama[1]

[1] Information Initiative Center
{munetomo,akama}@iic.hokudai.ac.jp
[2] Graduate School of Information Science and Technology,
Hokkaido University, Sapporo 060-0811, Japan
asim@uva.cims.hokudai.ac.jp

Abstract. This paper presents a problem-solving framework based on robust evolutionary search in GRID computing environment. Our problem-solving environment called *virtual innovative laboratory* performs simulator programs in parallel and optimize their input parameters employing a competent evolutionary algorithm with gene analysis. The objective of our project is to replace a part of human designer's try-and-error processes by a parallel and robust evolutionary search on GRID computing systems.

Keywords: GRID computing, problem-solving environment, evolutionary algorithms.

1 Introduction

In these days of high performance computers and parallel computation, general purpose optimization algorithms and frameworks, which can solve a wide range of problems using relatively large computational power, have emerged. Constructing a universal optimization framework is a hot topic in information science and technology. There exist some frameworks to solve optimization problems, e.g. NEOS[1,5,2] is an online server that can be used to submit optimization jobs over the Internet. NEOS includes some excellent solvers for linear programs and it also includes some good solvers for non-linear cases. There exist some professional software that also claims to handle a variety of optimization problems e.g. Palisade RISKOptimizer (www.palisade.com), Engineous iSIGHT (www.engineous.com), LINDO Systems (www.lindo.com).

This paper introduce yet another framework that exploits competent GAs and GRID computing to considerably reduce the development time by solving almost all kinds of optimization problems with a minimal input from the user. To the best of our knowledge there is no existing framework that employs competent genetic algorithms over a GRID environment to realize this kind of general purpose optimization framework.

For difficult problems like global optimization, heuristic search methods have been extensively studied; some of the best known types are simulated annealing,

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 473–480, 2007.
© Springer-Verlag Berlin Heidelberg 2007

tabu search, and genetic and evolutionary algorithms. These methods do not provide theoretical assurance to find an optimal solution. They sometimes find the best known solutions; however, which may be more than adequate for the task at hand[1]. In these kinds of algorithms, black-box optimization (BBO) algorithms get a great deal of attention, which require only inputs and outputs of the target objective functions. Even though genetic algorithms (GAs) are not BBO algorithms in a true sense yet they are employed in a lot of cases where we have very little domain knowledge is available[9]. A series of researches have been made to increase robustness by employing *gene analysis* to realize true BBO algorithms.

In this paper we have employed LINC (Linkage Identification by Nonlinearity Check)[7]-GA as a competent GA with gene analysis. LINC is a perturbation based linkage identification method. LINC seeks for nonlinearity in a pair of loci by bitwise perturbations to obtain linkage groups. Perturbation based methods are good at detecting sub-functions with small contribution to the over all fitness, but one of the disadvantages of perturbation based methods is that they require $O(l^2)$ fitness evaluations to detect linkage among a population, where l is the string length. Therefore perturbation methods are not feasible to use in environments with hard time constraints. This fact leads to parallel implementation of LINC-GA[8].

As discussed earlier, even though perturbation based methods have some advantages but they are computationally intensive, therefore we rely on a GRID computing environment to obtain the required computational resources. GRID computing is a form of distributed computing that involves coordinating and sharing computing, application, data, storage, or network resources across dynamic and geographically dispersed organizations. In the view of GRID computing, computing becomes pervasive and individual users (or client applications) gain access to computing resources (processors, storage, data, applications, and so on) as needed with little or no knowledge of where those resources are located or what the underlying technologies, hardware, operating system, and so on are[4]. Parallel implementation on GRID offers greater advantages over conventional parallel computing methods like cluster or super computers because we can perform fitness evaluations, gene analysis such as linkage identifications, and genetic optimization based on the analysis independently over a large number of computing nodes across different organizations. GRID also allows integrating data obtained from various resources not available locally. In order to make our project interoperable with other GRID systems we are using the globally recognized Globus Toolkit 4.0.2(GT4)[3].

This paper is organized as follows. Section 2 gives details about the algorithm and the architecture used in the proposed framework. Section 3 explains the experiments performed for benchmarking purposes and the results obtained. Section 4 is the conclusions section, in this section we also give some ideas for the future work.

2 Problem Solving Environment over the GRID

As discussed earlier our Virtual Innovative Laboratory (VIL) project seeks for realizing a virtual laboratory that *innovates* automatically to find optimal

solutions, design, etc. by employing robust evolutionary search based on GAs with gene analysis which analyzes target optimization problems automatically.

In this paper, we have constructed a web-service with automated gene analysis performed over a GRID computing environment to solve difficult optimization problems. The system we construct is composed of the following three components: (1) web-service interface, (2) job submit control working together with GRID middle-ware such as globus toolkit[3], and (3) robust evolutionary search that is a key engine for automated innovations. The following is the overview of parameter optimization employing the system.

1. The users prepare their simulators and XML input files that describe the specification of the problem to solve.
2. The VIL system automatically analyzes the problem based on the input files and samples of input-output relations generated by performing simulators. Analysis of the problem is realized indirectly by the gene analysis using LINC over the encoded strings for GAs.
3. The system searches optimal parameters by performing genetic operators based on the results of the gene analysis.

In the following, we explain the algorithm called parallel LINC employed in the VIL project, and show the detailed architecture of the system we propose.

2.1 Algorithm

The LINC checks nonlinearity for a pair of loci by bit-wise perturbations to obtain linkage groups, sets of loci that should be tightly linked in crossovers to avoid disruption of useful building blocks. We have developed a series of linkage identification techniques such as LINC, Linkage Identification with non-Monotonicity Detection (LIMD), Linkage Identification with Epistasis Measures (LIEM), etc. Another line of researches realizing robust evolutionary search are called estimation of distribution algorithms (EDAs), which analyzes the target problems by estimating probabilistic distributions from a set of promising solutions. Although we do not discuss the details here, we have developed another advanced method by combining the merits of both linkage identification and estimation of distribution.

In order to represent LINC algorithm mathematically let us consider a string $s = s_1 s_2 s_3 ... s_l$ that represents the target chromosome. Changes in fitness values by bit-wise perturbations to s are defined as follows:

$$\Delta f_i(s) = f(...\bar{s_i}.....) - f(...s_i.....) \tag{1}$$

$$\Delta f_j(s) = f(.....\bar{s_j}...) - f(.....s_j...) \tag{2}$$

$$\Delta f_{ij}(s) = f(...\bar{s_i}\bar{s_j}...) - f(...s_i s_j...) \tag{3}$$

Where, $\bar{s_i} = 1 - s_i$ and $\bar{s_j} = 1 - s_j$ in binary strings

Therefore, $\Delta f_{ij}(s) = \Delta f_i(s) + \Delta f_j(s)$ means that change in fitness value by perturbations on s_i and s_j is additive, which indicates a linear interaction

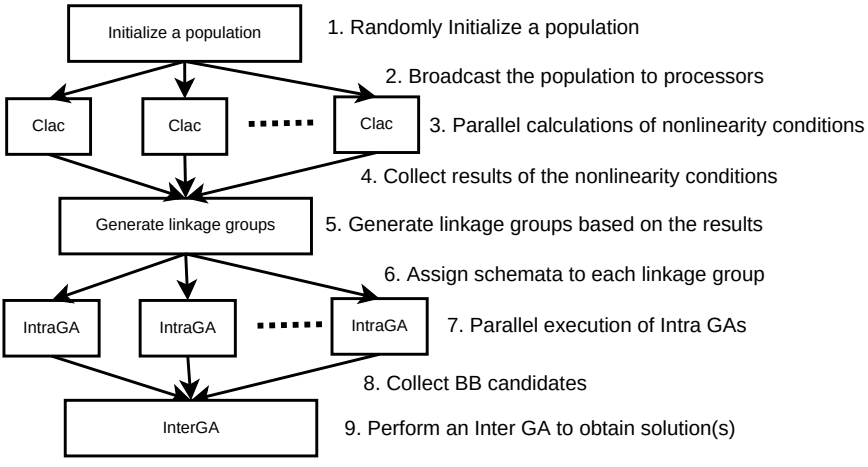


Fig. 1. pLINC Algorithm[8]

between the genes. On the other hand $\Delta f_{ij}(s) \neq \Delta f_i(s) + \Delta f_j(s)$ implies that they are not additive, which means non-linearity.

Figure 1 illustrates a parallel version of LINC(pLINC[8]) in algorithmic form. Note that computationally intensive linkage identification stage is parallelized.

2.2 Architecture

Figure 2 gives an overview of the overall architecture. After receiving the input and configuration data from the user the web service discovers the available resources in the virtual organization (VO) by sending a request to the root node of the VO. After this step the web service is able to run parallel jobs on all the available nodes in the VO that are running globus toolkits or any other job scheduler package like portable batch system (PBS).

Web Service. The implemented web service is running in the GRID container and there is only one running in the whole GRID. GT 4.0.x services used explicitly by our web service are GRAM (GRID Resource Allocation and Management) service and RFT (Reliable File Transfer) service. RFT service uses GRIDFTP for its internal working. GRAM service provides an interface for requesting and using remote system resources for the execution of *jobs* on remote hosts, while RFT is a web service that provides interface for controlling and monitoring third party file transfers.

Java Client. Client is a java application with graphical user interface that is actually consuming the web service using the WSDL (Web Service Definition

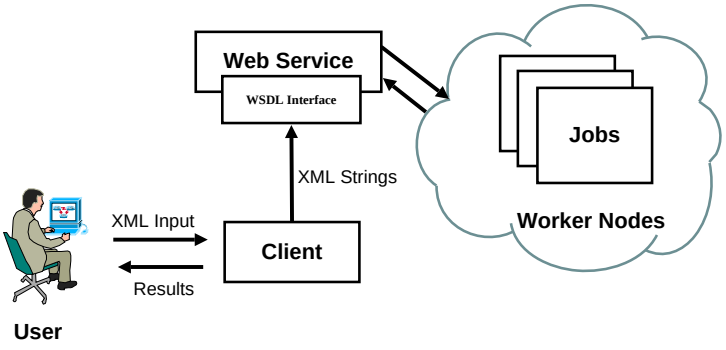


Fig. 2. Top level architecture of GA with linkage analysis over a GRID

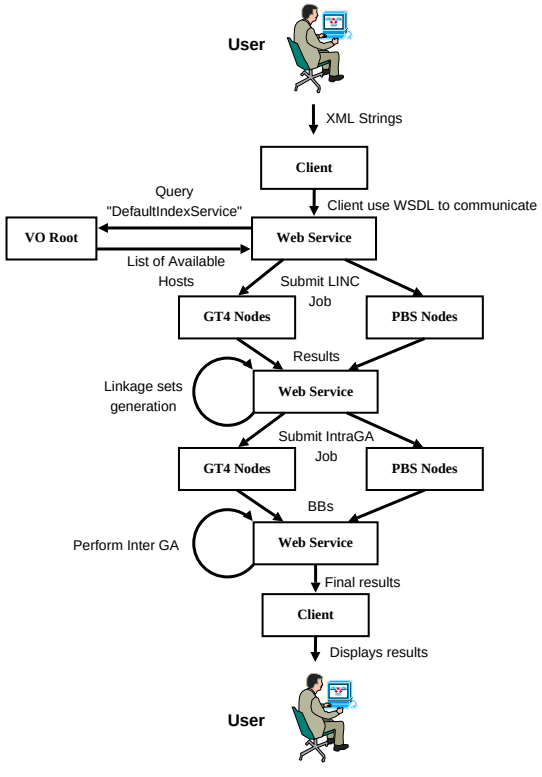


Fig. 3. Flow chart diagram displaying the overall sequence of events

Language) interface provided by the web service. For the time being we assume that there is only one Client running at one time in our experimental GRID, many clients can be running at the same time.

Table 1. Result of MAX-SAT problem solution employing pLINC

S no.	Problem no.	No. of Runs	Results over the Proposed Architecture			
			Clauses	Variables	Average Quality of Solution	Number of False Clauses for Each Run
0	uf20-010	5	91	20	1	0,0,0,0,0
1	uf20-0140	5	91	20	1	0,0,0,0,0
2	uf50-01000	5	218	50	0.97	6,8,7,6,7
3	uf50-0122	5	218	50	0.982	4,5,2,4,4
4	uf75-096	5	325	75	0.97	10,11,13,10,9
5	uuf100-01000	5	430	100	0.94	30,25,31,23,20
6	uuf100-01	5	430	100	0.95	25,20,19,24,22
7	uuf100-02	5	430	100	0.953	16,21,18,19,25
8	uf125-025	5	538	125	0.95	30,28,32,26,24
9	uf250-01	5	1065	250	0.94	75,63,69,76,35

Table 2. Execution time for running pLINC on different number of GRID nodes. (Time unit is seconds).

	Number of GRID Nodes		
	1	**2**	**3**
pLINC	1218	649	520
pLIMD	1289	679	486
pLIEM	1272	661	514

Worker Node. In this paper we call any node that has the potential of running the job created by the web service a worker node. In case of GT4 any node having GT4.0.x installed on it or any node running job scheduling software like PBS can become a worker node.

Figure 3 shows us a complete flow of data during the execution of the application. User is only required to input his data as an XML file which is handed over to the web service but the client application using the WSDL (Web Service Definition Language) interface provided by the web service. Web service is responsible to poll root node of the available VO to obtain a list of all the available nodes in that VO. After this step web service breaks down the application into smaller modules to execute it in parallel over the available worker nodes. The worker nodes calculate the linkage sets independently and return the results back to the web service node. Web service combines these inputs and after finishing sorting and selection it generates the final linkage sets that are necessary for the IntraGA step. Web service then break down the problem once again to assign task for IntraGA to each of the available worker nodes. These nodes give the result back to the web service. After combining the results the web service performs InterGA step and returns the result back to the client, which is then displayed to the user.

3 Experiment and Results

In this section we will discuss some experiments that we have performed and the results obtained by those experiments.

We employ the following test problems:

- Maximimum Satisfiability (MAX-SAT) problem.
- The sum of trap functions which is frequently employed to test GAs.

The MAX-SAT problem searches the maximum number of clauses which can be satisfied by any assignment. The problem comes into the category of NP-hard problems. Mathematically, MAX-SAT problem can be defined as follows:

$$f(s) = \sum_{i=1}^{N_c} f_i(s) \tag{4}$$

$$f_i(s) = w_i(s_j \vee s_{j+1} \vee ... \vee s_{j+N}) \tag{5}$$

Where, N_c is the number of clauses, N_v is one less then the number of variables in string S, w_i is the weight of the clause, $s_j, s_{j+1}, ..., s_{j+N}$ are a single bit or complement of that bit of the string S and $N \leq N_v$.

The sum of *Trap functions* is frequently employed to test evolutionary algorithms, which is defined as follows:

$$f(s) = \sum_{i=1}^{50} f_i(u_i) \tag{6}$$

$$f_i(u_i) = \begin{cases} 4 - u_i & if\ 0 \leq u_i \leq 4 \\ 5 & if\ u_i = 5 \end{cases} \tag{7}$$

Where u_i is number of ones in each 5-bit substring of s. This function is considered to be GA difficult and is often used for benchmarking purposes.

Table 1 shows the result of different SAT benchmarking problems taken from library provided by SATLIB[6]. It can be noticed from the "Average Quality of Solution" column that the quality of the solution decreases gradually with increase in the number of variables and clauses. Therefore, suggesting the robustness of the algorithms being used. Please note that the number of variables corresponds to the length of the chromosomes.

We have calculated the time taken by pLINC algorithm on an experimental GRID of 3 Heterogeneous nodes with Debian Linux as their OS. We have tested pLINC on the proposed architecture and we achieved good speedups as shown in table 2 although in a heterogeneous environment. A theoretical and empirical result on pLINC[8] indicates that speedups contine along the number of processors increases.

4 Conclusions

We have proposed a system on a GRID computing environment that enables the users to investigate their optimal design parameters automatically using genetic algorithms with gene analysis, only by giving the specification of their design problems and programming their simulation programs.

References

1. Czyzyk, J., Mesnier, M., More, J.: The NEOS server. IEEE Journal on Computational Science and Engineering 5, 68–75 (1998)
2. Dolan, E.: The NEOS server 4.0 administrative guide. Technical Memorandum ANL/MCS-TM-250, Mathematics and Computer Science Division, Argonne National Laboratory, May, Discusses the Server implementation and use in detail (2001)
3. Foster, I.: Globus toolkit version 4: Software for service-oriented systems. In: Jin, H., Reed, D., Jiang, W. (eds.) NPC 2005. LNCS, vol. 3779, pp. 2–13. Springer, Heidelberg (2005)
4. Foster, I., Kesselman, C.: The Grid: Blueprint for a New Computing Infrastructure. Morgan-Kaufman, San Francisco (1999)
5. Gropp, W., Mor'e, J.: Optimization environments and the neos server (1997)
6. Hoos, H.H., Stützle, T.: SATLIB: An Online Resource for Research on SAT. In: pp. 283–292.
7. Munetomo, M., Goldberg, D.E.: Identifying linkage groups by nonlinearity/nonmonotonicity detection. In: Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M., Smith, R.E. (eds.) Proceedings of the Genetic and Evolutionary Computation Conference, vol. 1, pp. 433–440. Morgan Kaufmann, Orlando, Florida, USA (1999)
8. Munetomo, M., Murao, N., Akama, K.: A parallel genetic algorithm based on linkage identification. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 1222–1233. Springer, Heidelberg (2003)
9. Tsuji, M.: Designing Genetic Algorithm Based on Exploration and Exploitation of Gene Linkage. PhD thesis, Hokkaido University, Sapporo, Japan (2007)

A Resources Virtualization Approach Supporting Uniform Access to Heterogeneous Grid Resources*

Cunhao Fang[1], Yaoxue Zhang[2], and Song Cao[3]

[1] Tsinghua National Labatory of Inforamation Science and Technology
[2] Department of Computer Science and Technology,Tsinghua Univeristy
[3] School of Software, Tsinghua University
100084 Beijing, P.R. China
fangch@tsinghua.edu.cn

1 Introduction

Grid system has various kinds of resources such as computing resources, storage resources, instrument resources, data resources, etc. However, because of the differences of formats, descriptions, structures, and access modes of these resources, grid computing fails to access these resources uniformly and make full use of them. How to organize and manage all sorts of resources as a whole In Grid Environment, and provide the upper application with coherent description as well as uniform access interface is the problem of coherency access of different resources In Grid Environment.

Resources virtualization makes better use of the dynamic and distributed service resources under grid environment. It is a proper method to solve the problem described above.

Web service is a set of protocol warehouse defined by XML. Through protocols and standards such as SOAP, WSDL, UDDI, WSFL, BPEL4WS, it provides uniform service registry, discovery, binding, and integration mechanism facing Internet. Web service has become an important mechanism of resources mutual manipulation underlying extensive environment.

This research starts with various physical resources In Grid Environment, transforms and encapsulates various resources into services on the basis of Web service technology, and further provides upper applications with uniform service resources through uniform description and management of these services. Based on this, multi application cases with the same function body are merged, different function bodies are analyzed, and services resource view on the user function layer is extracted in support of service logic integration process on the user end. The whole virtualization architecture consists of three ***Views*** and three ***Processes***, as shown in Fig.1.

The remainder of the paper is organized as follows. Related definitions are presented in Section 2. Section 3 introduces in detail the process of resources virtualization and organization, followed by conclusion in Section 4.

* This paper is sponsored by National Natural Science Foundation of China (NO. 90604027).

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 481–487, 2007.
© Springer-Verlag Berlin Heidelberg 2007

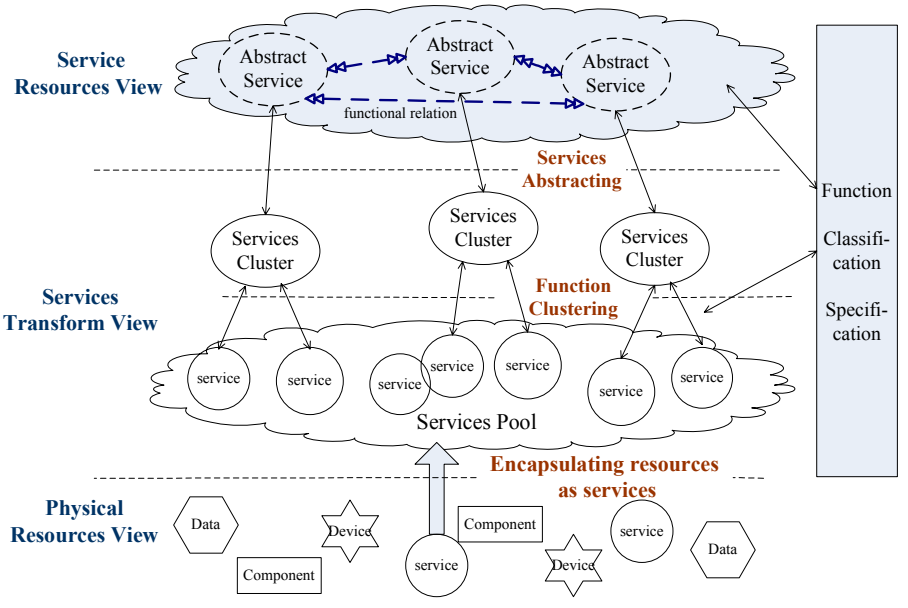


Fig. 1. Layered Architecture of the Framework

2 Some Definitions

Grid system has various kinds of resources such as computing resources, storage resources, instrument resources, data resources, etc. R refer to the resources pool, all resources in the grid system are in it. Suppose there are n kinds of resources in the resources pool, they are R1, R2, …, Rn.

Definition 1

Resource $\boldsymbol{R_i}$ $\boldsymbol{=(Category, F_i, I_i, P_i)}$

Category is the kind of resource $\boldsymbol{R_i}$;

$\boldsymbol{F_i}$ is the functional description of resource $\boldsymbol{R_i}$;

$\boldsymbol{I_i}$ is the interface of resource $\boldsymbol{R_i}$ (such as calling parameter and running results, etc.);

$\boldsymbol{P_i}$ indicates the using policy set of resource $\boldsymbol{R_i}$.

We use deputy mechanism to demonstrate different kinds of services through service uniformly. With regard to services, we have the following definition:

Definition 2

Service $\boldsymbol{S_i}$ = $\boldsymbol{(Name_i, BaseInfo_i, Interface_i, Binding_i, Function_i, AD_i)}$

$Name_i$ is service identification;

$BaseInfo_i$ is basic non-behavior information of the service including name, author, version, manufacture, etc;

Interface$_i$ is the interface of service $\boldsymbol{S_i}$ (such as calling parameter and running results, etc.);

Binding$_i$ is binding protocol set of the service;

Function$_i$ is function description of the service;

AD$_i$ is application domain of the service.

3 Resources Virtualization and Organization Process

As Fig. 1 shows, resources virtualization and organization composes of three ***Views*** and three ***Processes***:

Three ***Views:***

1) Physical Resources View: this view refers to various Grid resources and corresponding supporting tools (e.g. resource deploying tools).

2) Services Transform View: this view aims to implement the uniform resources transform (to Web Services format) and to organize the structure of transformed ***Services Pool*** by analyzing, categorizing and managing these service resources according to ***Function Classification Specification (FCS)***.

3) Service Resource View: this view aims to provide users with well-organized services resource views on the application level through further mining the semantic relations of services, to support service active discovery, service composition on demand and other upper-layer applications.

Three ***Processes:***

1) Service instancing process of Resources (Encapsulating resources as services)***:***

As described above, Web service based on Internet protocol provides us an important mechanism to realize resources mutual manipulation underlying extensive environment. Thus, the first step of resources virtualization and organization process is to transform all resources into services and pack them up in the service format. Through uniform description and management of these services, uniform service resources are provided to upper application.

In this process, we adopt service component deputy mechanism. During the active service implementation process, we view various resource formats such as components, data and equipments as service component deputies. Transferring between these resources is converted into service requests and responds between service component deputies (based on message mechanism).

Resource $\boldsymbol{R_i}$ is transformed and encapsulated as a web services (denoted as $\boldsymbol{S_i}$) by mapping the information of $\boldsymbol{R_i}$ to Web Services Specification.(e.g. mapping the $\boldsymbol{R_i.F_i}$ to the UDDI registration information of service $\boldsymbol{S_i}$, mapping the $\boldsymbol{R_i.I_i}$ to the WSDL description information, mapping the $\boldsymbol{R_i.P_i}$ to the WSDL binding information, etc.). After that, a ***Service Agent*** is initiated to delegate the "virtual service $\boldsymbol{S_i}$". the ***Service Agent*** first deploy the $\boldsymbol{R_i}$ entity into the grid environment according to the resource kind denoted by $\boldsymbol{R_i.Category}$, when a calling request comes, the ***Service Agent*** calls resource $\boldsymbol{R_i}$ and encapsulate the calling results of resource entity to the message style of services $\boldsymbol{S_i}$. Through the description mapping and ***Service Agent*** mechanism, resource $\boldsymbol{R_i}$ is completely transformed to service $\boldsymbol{S_i}$.

We present the general process of resources service instancing as follows. With regard to service resources, we directly deploy them into the grid environment; with regard to other resources, we can firstly deploy them into the grid environment through deputy mechanism, and then pack them up into services.

$\forall r_i \in R_{set}.1 \le i \le n$

If $r_i \in WS$ then $r_i \rightarrow R_{set}$

Else

$GridServices(r_i \vee deploy(r_i)) \rightarrow R_{set}$

Endif

2) Function Clustering Process of Services:

Service function clustering is to extract the functions of all service components, and to constitute a function categorization norm according to the relationship between different functions.

Based on the function categorization norm, a set of services with similar functions are merged upon this function, and is demonstrated as an abstract service. There are some related definitions:

(1) ***Func(S_i)*** is the function extraction function, indicating the extracted function of service S_i.

(2) ***FS*** is the function categorization norm. Initially, it is empty. According to each service, through function extraction function, extracted function key words are made one item of FS. Users are required to define relationships between different function key words.

The service clustering flow is described as followings. (a) Services pool directory according to the ***FCS*** is setup. Each service will advertise its function in the directory. (b) Based on the Services pool directory, a set of services with similar functions are merged upon a directory item, and are demonstrated as a service cluster. (c) The whole service cluster forms a net-shaped organization structure, which sets the foundation for service discovery and further organization.

The general process of service function clustering is described as follows:

$\forall s_i \in S_{set}.1 \le i \le n$

If $(\exists funcItem_k \in FS, map(Func(s_i), funcItem_k) = TRUE)$

Then

$ServiceCluster(s_i, funcItem_k)$;

Else

$New(funcItem, m+1)$;

$funcItem_{m+1} = Func(s_i)$;

$ServiceCluster(s_i, funcItem_k)$

Endif

3) Service Abstracting and Functional Relations linking Process:

After the above two processes, a set of services with similar functions are linked to the same function item with identical transferring interfaces and semantics transferring relationships.

The relationships between services can be combination, inheritance, inclusion, equation (replacement), calling, etc. Combinations compose of gradation, filiations, coalition, recursion, etc. No matter the relationship is combination, inheritance, inclusion, equation, or transferring, there are different semantic relationships between these services.

In order to better virtualize service resources, we organize up the semantic relationships between services to form up a hierarchical function relationship network, namely service function semantics Ontology and form the services resources view on the application semantics level. There are some related definitions.

(1)Abstract function

$$AS(\ funcItem_k\ ) = \bigcup s_i, \forall s_i, map(Func(s_i),\ funcItem_k\ ) = TRUE$$

(2)Semantic relationships between abstract services.

Let Si and Sj be two different sub services in A1, the relationships between them R has the same logic relationship with function sub set defined in Def 4.

(3) Service function semantics Ontology

Service function semantics ontology composes of three parts: a set of abstract services, integration combination relationships between services, and information sequences of integration.

Ontology = （$\boldsymbol{A_{AS\text{-}subset}}$， $\boldsymbol{A_{Message}}$， $\boldsymbol{A_{Scenario}}$）

• $\boldsymbol{A_{AS\text{-}subet}}$ =($\boldsymbol{AS_1, AS_2, \ldots, AS_n}$) It is the abstract services subset cited in the ultimate active service combination layer. This active service is composed of services in this subset.

• $\boldsymbol{A_{Message}}$ =($\boldsymbol{M_1, M_2, \ldots, M_n}$) is a finite set of messages. One message is composed of message type, message dispatcher, message receiver, and message body. Message body includes message parameters and protocol-related data. Definitions of this part is the same as the message definition of WSDL.

• $\boldsymbol{A_{Scenario}}$ is the composition scheme of active service defined on abstract service component subsets $\boldsymbol{AS_1}$ and $\boldsymbol{AS_2}$. It describes sequence relationships and controlling relationships of each abstract service in AS.

On the bases of the above definitions, the definition process of semantics relationships between abstract services are defined as follows:

$\forall as_i \in AS_{set.} \exists funcItem_i, as_i = AS(\ funcItem_i), 1 \le i \le n,$

$\forall as_i, as_j \in AS_{set,}\ 1 \le i, j \le n$

If （$Relation(as_i, as_j) = TRUE$）

Then

Add（$as_i, as_j, Relation(as_i, as_j), Message(as_i, as_j)$） to ***Ontology***）

Endif

With the help of service function semantics Ontology, we can further record users' requirement analysis and re-composition results after functions decomposition. We use service function semantics Ontology to define the relationships between this the sub-functions of decomposed service.

4 Conclusion

In this paper, we present a virtualization architecture consisting of three views and three processes to organize and manage all sorts of resources as a whole In Grid Environment, and provide the upper application with coherent description as well as uniform access interface.

First, various physical resources In Grid Environment are transformed and encapsulated into services on the basis of Web service technology, and uniform service resources are provided for upper applications through uniform description and management of these services. Then by analyzing, identifying, classifying, organizing, storing, and managing these transformed services (i.e. Function Clustering Process of Services), multi application cases with the same function body are merged, and then the semantic relationships between services are further mined to better virtualize service resources. At last, we organize the semantic relations among services to establish a hierarchical function relationship network, and form the Services Resources View on the application semantics level.

References

1. yaoxue, Z., cunhao, F.: Active services: Concepts, Architecture and Implementation [M]. Thomson Learning (2005)
2. Borenstein, N., Freed, N.: MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies, RFC 1521 (September 1993)
3. Chan, S.W.K., Franklin, J.: Dynamic context generation for natural language understanding: a multifaceted knowledge approach. IEEE Transactions on Systems, Man and Cybernetics, Part A 3(31), 23–41 (2003)
4. Hristidis, V., Papakonstantinou, Y., Balmin, A.: Keyword proximity search on XML graphs. In: Proc. of the 19th International Conference on Data Engineering, 5–8 March, pp. 367–378 (2003)
5. Lawrence, S., Giles, C.L., Fong, S.: Natural language grammatical inference with recurrent neural networks. IEEE Transactions on Knowledge and Data Engineering 12(1), 126–140 (2000)
6. Meuller, A., Mundt, T., Lindner, W.: Using XML to semi-automatically derive user interfaces. Second International Workshop on User Interfaces to Data Intensive Systems (May 31 to June 01, 2001)
7. Nakauchi, K., Ishikawa, Y., Morikawa, H., Aoyama, T.: Peer-to-peer keyword search using keyword relationship. In: Proc. of the CCGrid 2003, pp. 359–366. ACM Press, New York (2003)

8. Hai, Z.G.: A problem-oriented and rule-based component repository. The Journal of Systems and Software 50, 201–208 (2000)
9. Fang, C.H., Zhang, Y.X., Xu, K.G.: An XML-based data communication solution for program mining. Intelligent Data Engineering and Automated Learning, 4th International Conference, pp. 569–575. Springer, Heidelberg (2003)
10. Zhang, Y.X., Fang, C.H., Wang, Y.: A feedback-driven online scheduler for processes with imprecise computing. Journal of Software 15(4), 616–623 (2004)

Pattern-Based Verification for Trees*

Milan Češka, Pavel Erlebach, and Tomáš Vojnar

FIT, Brno University of Technology, Božetěchova 2, CZ-61266, Brno, Czech Republic
{ceska,erlebach,vojnar}@fit.vutbr.cz

Abstract. Pattern-based verification trying to abstract away the concrete number of repeated memory structures is one of the approaches that have recently been proposed for verification of programs using dynamic data structures linked with pointers. It proved to be very efficient and promising on extended linear data structures. In this paper, we overview some possibilities how to extend this approach to programs over tree structures.

1 Introduction

This paper addresses the problem of formal verification of *programs manipulating dynamic data structures linked by pointers* (such as lists, trees, etc.). In such programs, many mistakes can be easily made since the source code is usually not very transparent, and the functionality of the program is not apparent at the first sight. Consequently, a possibility of proving correctness of such programs is highly desirable. However, the verification of these programs is also quite complex as (1) they are infinite-state due to working with unbounded data structures, and (2) the objects that are manipulated here are in general unrestricted graphs.

The research on formal verification of programs with dynamic data structures is nowadays quite live, and there have appeared many different approaches in this area, such as [10,11,1,5,2,9], differing in their formal roots, degree of automation, and the kind of program data structures and properties they can verify. Among the recently proposed methods for verification of programs with dynamic data structures there is also the so-called *pattern-based verification* [12,4]. This method is based on detecting repeated adjacent subgraphs in heap graphs and collapsing them into a single summary occurrence (thus ignoring their precise number). The method has especially been studied in the context of verifying extended linear data structures, i.e. data structures with a linear skeleton and possibly some additional edges on top of it. Such structures, including non-circular as well as circular singly-linked and doubly-linked lists, possibly with additional pointers to the head, tail, etc., are very common in practice. Pattern-based verification on these structures has been made fully automated [4,8,7,3]. The method can verify properties like absence of null pointer dereferences, dangling pointers, absence of garbage, but also more complex safety properties (e.g. shape invariance).

* This work was supported by the Czech Grant Agency within projects 102/05/H050 and 102/07/0322, by the Czech-French project Barrande 2-06-27, and by the Czech Ministry of Education under the project *Security-Oriented Research in Information Technology*, contract CEZ MSM 0021630528.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 488–496, 2007.
© Springer-Verlag Berlin Heidelberg 2007

On the above described kind of extended linear structures, pattern-based verification turned out to be quite efficient. A natural question is then whether and how it can be extended to programs over tree-like structures. That is why, in this paper, we discuss the problems that appear in pattern-based verification when one proceeds from linear structures to tree-like ones (including tree structures with parent pointers and perhaps other additional pointers, such as root pointers, pointers to separately allocated data nodes, etc.). These problems include a need to work with a different kind of patterns and also a state space explosion problem that appears when one tries to straightforwardly transform the methods from the linear setting into the tree setting. We outline a way of how to cope with this state explosion. The results we describe do not yet provide a fully satisfying solution, but provide a significant and promising improvement over the trivial generalisation of pattern-based verification from linear structures to trees. Due to space limitations, the paper provides just an overview of the results—more details can be found in [6].

Plan of the Paper. The paper is organized as follows. In Section 2, the main principle of the original linear variant of pattern-based verification is briefly recalled. In Section 3, the extension of the method to trees is discussed. In Section 4, some early experimental results are presented. Finally, we conclude by a discussion of the achieved results in Section 5.

2 Pattern-Based Verification

In pattern-based verification, sets of memory configurations are represented as *abstract shape graphs* (ASGs) which are abstract memory graphs with basically two types of nodes: simple nodes corresponding to particular concrete nodes allocated in the memory (a list node, a tree node, etc.) and *summary nodes* behind which two or more "similar" concrete nodes are hidden (the exact number is abstracted away—note that such nodes cannot be, e.g., pointed by a pointer variable as this would clearly make them distinguishable from each other). The simple nodes are labelled with pointer variables pointing to them whereas summary nodes are labelled by the associated *pattern*. The pattern represents a memory subgraph whose two or more adjacent, mutually interconnected *instances* (or occurrences) are hidden behind the summary node. The nodes are linked by edges corresponding to the pointer links binding them in the memory and are labelled by the selectors (like `next`, `previous`, `left`, `right`, etc.) to which the appropriate pointer links correspond.

When verifying a program within pattern-based verification, one iteratively computes the set of all ASGs reachable at each program line. Particular program statements are performed on ASGs in three phases: (1) The given ASG is first (partially) *materialised* if the manipulation would cause a pointer variable point to a summary node (e.g., via a statement like `x = y->next`), which must not happen as the target of the pointer variable in such a case would not be well defined. That is why one concrete occurrence of the appropriate pattern hidden behind the summary node is explicitly instantiated and then used. (2) The statement is performed on concrete memory nodes as usual. (3) The resulting graph is *summarised*, i.e. searched for repeated adjacent occurrence of patterns that are then merged into a single summary node.

The idea of pattern-based verification has first appeared in [12] in a semi-automated framework whose user had to supply not only the input configurations and the program to be verified, but also the patterns to be used for abstraction. In the work, linear structures extended with possibly additional pointers to some shared nodes (like singly-linked or doubly-linked lists with head/tail pointers) were considered. The approach was able to handle only a single pattern which restricted its applicability to dynamic structures with a simple, fixed inner structure. In [4,8,7,3], several extensions were introduced which made pattern-based verification over extended linear structures fully automated (by providing an automated discovery of patterns) and enhanced its generality (through a more general definition of a pattern and a possibility to work with more than one pattern).

In the rest of the section, we will say a bit more on the principles of automated pattern-based verification on extended linear structures in order to be able to contrast them with the tree case presented in Section 3.

2.1 The Main Principles of Linear Pattern-Based Verification

Memory patterns considered in [4] for extended linear structures are defined as a 5-tuple $P = (N^P, e^P, x^P, S^P, E^P)$ where N^P is a set of nodes, $e^P \in N^P$ is an entry node, x^P is an exit node ($e^P \neq x^P$), $S^P \subset N^P$ is a set of the so-called *shared* nodes, and $E^P \subseteq N^P \times Sel \times N^P$ is a set of edges of the memory pattern (labelled by pointer selectors). The entry and exit nodes delimit the linear skeleton of the pattern and represent the main connection points of instances of the pattern in concrete memory graphs to their surroundings. The shared nodes are shared among all instances of the patterns (like the head element of a list pointed to by head pointers) and play a somewhat similar role as global variables in programs with recursive procedures. The remaining nodes are internal to the pattern and thus also to its instances.

An *automatic discovery of patterns* is activated before every summarisation attempt (in [4], this holds only till some pattern is found since [4] is still restricted to a single pattern; this restriction is relaxed in [3]). The discovery of patterns consists in exploring all nodes of encountered shape graphs and searching for subgraphs that can be delimited by an entry and exit node (and perhaps shared nodes) and that appears at least twice in the given shape graph (before and after its original occurrence). For illustration, the pattern detected when working with singly-linked lists (with data abstracted away) is $P_1 = (\{e,x\}, e, x, \{\}, \{(e, \texttt{next}, x)\})$. The pattern discovered in doubly-linked lists with data stored in separately allocated nodes and with tail pointers is $P_2 = (\{e,x,t,d\}, e, x, \{t\}, \{(e, \texttt{next}, x), (x, \texttt{prev}, e), (e, \texttt{data}, d), (e, \texttt{tail}, t), (x, \texttt{tail}, t)\}$.

Summarisation consists in searching a given shape graph for subgraphs isomorphic with known patterns. If at least two adjacent instances of some patterns are found (linked via the exit/entry nodes), all nodes belonging to these instances are replaced with a single summary node (apart from the exit node of the last instance). Some further restrictions apply ensuring correctness and reversibility of the summarisation—e.g., no program variable can point to any node of any of the detected instances of the pattern.

We then distinguish a complete and a partial *materialisation*. In a complete materialisation, the summary node is replaced by two new instances of the pattern, while in

a partial materialisation, one new instance of the pattern is inserted into the graph (followed by the summary node). The former case corresponds to having just two instances of the pattern hidden behind the summary node whereas the latter to having three or more. Both of these variants must always be explored as we do not record the exact number of summarised instances. In the case of a partial materialisation, we further distinguish a forward and a backward materialisation depending on the mutual positioning of the preserved summary node and the materialised instance of the pattern.

3 Pattern-Based Verification on Trees

The main contribution of this paper is an extension of pattern-based verification to programs manipulating *tree structures*. This extension implies changes in all parts of the method and also a need for some optimisations as a serious state space explosion problem appears. Due to space restrictions, all the changes are only briefly outlined in the section, a detailed description together with formal definitions can be found in [6].

3.1 Patterns and Their Discovery

While in the linear variant of pattern-based verification there was just one exit node in a memory pattern, for trees there must be more exit nodes to allow us to cope with the branching of the structure. So, now, a pattern is a tuple $P = (N^P, e^P, X^P, S^P, E^P)$ where $X^P \subset N^P$ is *the set of exit nodes* and other symbols keep their original meaning. The pattern of a binary tree would have two exit nodes, in case of ternary trees, there will be three exit nodes, etc.

The algorithm of discovering patterns is very similar to the linear case with the difference that it is needed to recognise and distinguish the *parent* nodes. While in the linear case, it was not necessary to distinguish predecessors and successors (and, e.g., in the case of doubly-linked lists, this would even lack sense as predecessors and successors are structurally indistinguishable), in trees, the role of a parent is much different than that of successors. Its identification is needed in order for the search of patterns to be restricted to the nodes below the parent only. The reason is illustrated in Fig. 1. For simplicity, the names of selectors are omitted. In Fig. 1(a), a pattern that we would like to be discovered in common binary trees with parent pointers is shown. Entry and exit nodes are named as e and x_1, x_2, respectively. A repeated materialisation of the pattern from a single summary node leads to the structure shown in Fig. 1(b). However, if we do not distinguish the role of parent nodes, in such trees, we can also

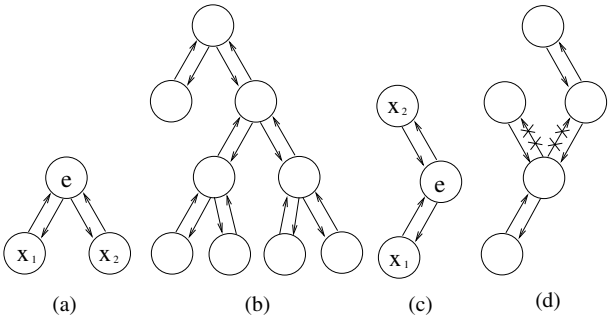


Fig. 1. Patterns in binary trees with parent pointers

detect the undesirable pattern shown in Fig. 1(c). If we tried to materialise this pattern repeatedly, we would obtain the structure shown in Fig. 1(d) which is not sensible as there would be two edges (marked with crosses) with the same selector leading out of a single node.

3.2 Summarisation

A major change in summarisation in pattern-based verification on trees comes from the fact that trees of different sizes have different numbers of leaves. While summarisation of any (long enough) linear structure results in one and the same abstract shape graph with one exit node (the end of a list), summarisation of two trees of a different size, if done in the same way as for linear structures, results in different abstract structures that differ in the number of exit nodes (i.e. leaves), which are not summarised. These exit nodes together with the graphs rooted at them will be further called *tails*. Note that tails may really be subgraphs more complex than a single node: For example, they can have the form of a node with a self-loop, a node with an attached data node, a node pointed by a program variable, a beginning of a linear list attached to a tree, or generally an arbitrary subgraph that cannot be summarised.

To cope with the above, we propose a two-phase summarisation for trees. The first phase is equal to summarisation on linear structures (up to minor changes related to a re-definition of the pattern). The second phase consists in clustering the same tails into the so-called *multi-tails* which represent *two or more equal tails*. The heart of the second phase of the summarisation is the equivalence relation on the tails. Let us outline the main idea of the equivalence here—its formal definition is rather technical and can be found in [6]. Two tails are equivalent iff they are both linked to the same summary node in the same way (i.e. via the same edges in both directions), they are isomorphic, and no pointer variable points to any node of any of the tails (tails with a program variable pointing into them are always unique).

After the first phase of summarisation, the tails are partitioned according to the described equivalence. Unique tails are then left unchanged, while the others are treated in the following way. From every non-singleton set of the partition, one random representative is chosen and kept in the shape graph (just the type of the edges leading to it is changed to a special value specific for multi-tails), and the other tails are deleted. In this way, it is ensured that arbitrary sets of reachable trees of arbitrary forms will always summarise to a finite number of (finite) abstract shape graphs.

3.3 Materialisation

The changes in materialisation when using patterns over trees are the most complex and are related mainly to the multi-tail concept and to the increased number of exit nodes. Materialisation of a summary node with a left and a right multi-tail, which represents the simplest binary tree structure (and which we consider as an illustration example in the rest of the section), results in over 40 new shape graphs. Why such a high number?

Like in the linear case of a partial materialisation, a new, concrete instance of a pattern is inserted into the place of the original summary node. However, as the pattern has multiple exit nodes, its materialised instance will not be followed by a single new

summary node as in the linear case, but by several new summary nodes (two for binary trees) where each of them represents a subtree of the materialised node. Each of these new summary nodes can have various combinations of tails and multi-tails since it is needed to cover all possible irregularities of subtrees of the materialised node. For example, in the materialisation mentioned above, the new summary nodes may have two multi-tails (the subtrees rooted at them are again proper trees with at least two left and two right tails), or a left multi-tail and a right tail (the subtrees have two or more left tails, but only one right tail), or just a right multi-tail (the subtrees have at least two right tails and no left tails), etc. This is the main reason of the high number of the resulting shape graphs.

Moreover, analogously to a complete materialisation in the linear case, one has to also consider replacing the new summary nodes by a concrete instance of the pattern. Finally, we have to even consider a new possibility specific for the tree setting, namely, the possibility of completely omitting the new summary node: If there are only two instances of a pattern hidden behind a summary node, one of them takes the place of the summary node, one goes to one of the branches, and the other branch is left empty).

Let us note that not all combinations of tails and multi-tails arising as described above are admissible—e.g., if both new summary nodes possible in the discussed example are replaced with single concrete nodes, the condition that *one multi-tail represents two or more tails* will not hold. In the original abstract shape graph there was a left multi-tail, and so in all resulting graphs there must be at least one left multi-tail, or at least two left tails. Otherwise, some left successors are lost. The same holds for the right tails too.

As we already mentioned, materialisation of the simplest tree structure results in over 40 new shape graphs. In the case of one additional pointer pointing to one of the tails, the number gets almost 100, and in the case of ternary trees with one additional pointer, it becomes over 2000 structures resulting from one materialisation operation. This would make the verification infeasible, and so some optimisations are needed.

3.4 Optimisations of the Materialisation

The reason of the state explosion within the basic materialisation on trees mentioned above lays in a too precise representation of the various irregularities of trees. But, our experiments show that in practical programs, it is usually not necessary to distinguish between, e.g., a tree with one right leaf (i.e. a leaf that is the right son of its father) and a tree with two right leaves, which is, however, a distinction that we enforce by defining multi-tails as covering two or more tails.

If we change the understanding of a multi-tail from *two or more tails* to *one or more tails*, we achieve a significant reduction of the materialisation state space. A materialisation of a summary node with two multi-tails (the same case that we used as an illustrative example in Section 3.3) would then not result in shape graphs where the new summary nodes can have normal tails (e.g. a summary node with a left tail and a right multi-tail is fully covered with a more general summary node with two multi-tails since this summary node now represents a subtree with one or more left and one or more right tail). The number of the resulting shape graphs is reduced to 10.

How would the situation look like if we continue in the above direction and set the meaning of the multi-tail to *zero or more tails*? A materialisation of the summary node

with two multi-tails that we considered above would result in only two structures. In the first resulting shape graph, the materialised summary node would be replaced by an instance of the pattern where all exit nodes are summary nodes with two multi-tails (an analogy to the partial materialisation), and in the second resulting graph, the materialised summary node would be replaced by a single tail (an analogy to the complete materialisation). There would not be needed any combinations of tails and multi-tails of summary nodes since a new summary node with two multi-tails would cover all possibilities of concrete subtrees (even the irregular ones like a tree reduced to a list and so on). Thanks to this reduction of the materialisation state space, the verification time is cut from hours or days down to seconds or minutes in the examples we consider in Section 4. The optimisations cause a slight increase of imprecision of the abstraction, but in the practical examples we considered, it does not make any difference.

4 Experiments

After the optimisations mentioned in the previous section, the verification times become reasonable for most library procedures manipulating binary trees. The times in seconds (if not explicitly declared otherwise) that we obtained from an initial implementation of our method are shown in Table 1. The prototype was implemented in SWI Prolog, and the tests were run on a PC with an AMD Athlon 2GHz processor.

The procedures mentioned in Table 1 are the following: `search` performs a random search in a tree (as the data contents is abstracted). The `deleteAll(*)` procedures delete all nodes of a tree, for which `deleteAll` exploits the parent pointers, while `deleteAll*` does not—in its case, the deletion consists in a repeated deletion of the leftmost leaf (repeatedly searched starting from the root). The `insertLast` and `deleteLast` procedures insert and delete a random leaf, respectively. The `tree2list` procedure converts a tree to the linear list via the postfix traversal. The `insert` and `delete` procedures insert and delete random node into/from a tree, respectively. Finally, `DSWtraversal` performs the Deutsch-Schorr-Waite tree traversal, namely the Lindstrom variant.

Table 1. Verification times for some procedures manipulating trees (in seconds if not stated otherwise)

procedure	Binary trees	Binary trees with parents
search	1.22	2.26
deleteAll	-	1.80
deleteAll*	4.26	6.58
insertLast	1.42	2.64
deleteLast	8.00	12.30
tree2list	52.40	77.44
insert	6.00	12.30
delete	459.2	840.7
DSWtraversal	2.6h	5.1h

In all the cases, we automatically verified that no null dereference and no memory leakage can occur. Moreover, using the generated reachable abstract shape graphs, we were able to manually verify various other safety properties (such as shape invariance, etc.). Let us note that the time needed to verify the `delete` procedure was higher due to the nodes manipulations used when a node with both children is deleted. The rightmost leaf of the left subtree has to be found, exchanged with the node which should be deleted, and then the node can be deleted as a leaf. The verification of `DSWtraversal`

needed even more time because of its vast abstract state space caused by its use of four program variables pointing to the tree with a large number of combinations of positions of these variables in the tree.

5 Conclusion

In this paper, we briefly introduced an extension of the pattern-based verification method from programs over extended linear data structures to programs over tree structures, including tree structures with parent pointers and perhaps some other additional pointers (like root pointers, pointers to separate data nodes, etc.). We have especially discussed the state explosion in the materialisation (i.e. concretisation) step, which occurs in this setting due to having to deal with all various irregularities of trees, and ways how to deal with this problem.

The verification times that we obtain from our early prototype implementation are one to two orders of magnitude higher than when handling linear structures. They are still not yet fully satisfactory, but they are several orders of magnitude better than when using a straightforward extension of the principles of pattern-based verification from linear structures to trees. Other existing tools capable of handling programs over trees can sometimes provide better verification times, but are often less general (like the grammar-based shape analysis [9]) or less automated (like PALE based on the WSkS logic and tree automata [10] or TVLA based on first-order predicate logic with transitive closure [11]).

The remaining efficiency problem of the analysis is that it still preserves relatively a lot of information about the various irregularities that may arise in trees. If pattern-based verification is to achieve similarly nice results on trees as on extended linear structures, some further optimisations are still needed, perhaps in the form of some sort of a counterexample-guided abstraction refinement loop allowing one to drop more information about the structure and then reclaim it on demand.

Interestingly, the feature of our analysis of keeping a relatively precise information about the structures handled could become an advantage if the analysis was applied to a dynamic data structure with a complicated internal scheme (e.g. trees whose nodes have an attached linked substructure of a fixed form such as a circular list with four nodes, etc.). Such structures could be handled by our analysis with no additional need of a manual intervention or a dramatic increase of the verification time.

References

1. Bouajjani, A., Habermehl, P., Moro, P., Vojnar, T.: Verifying Programs with Dynamic 1-Selector-Linked Structures in Regular Model Checking. In: Halbwachs, N., Zuck, L.D. (eds.) TACAS 2005. LNCS, vol. 3440, Springer, Heidelberg (2005)
2. Bouajjani, A., Habermehl, P., Rogalewicz, A., Vojnar, T.: Abstract Regular Tree Model Checking of Complex Dynamic Data Structures. In: Yi, K. (ed.) SAS 2006. LNCS, vol. 4134, Springer, Heidelberg (2006)
3. Češka, M., Erlebach, P., Vojnar, T.: Generalized Multi-Pattern-Based Verification of Programs with Linear Linked Structures. Accepted to Formal Aspect of Computing. Springer, Heidelberg (2006)

4. Češka, M., Erlebach, P., Vojnar, T.: Pattern-Based Verification of Programs with Extended Linear Linked Data Structures. In: Proc. of AVOCS'05. ENTCS, vol. 145, pp. 113–130 (2006)
5. Distefano, D., O'Hearn, P.W., Yang, H.: A Local Shape Analysis Based on Separation Logic. In: Hermanns, H., Palsberg, J. (eds.) TACAS 2006 and ETAPS 2006. LNCS, vol. 3920, Springer, Heidelberg (2006)
6. Erlebach, P.: Automated Formal Verification of Programs Working with Dynamic Data Structures. PhD. thesis, Brno University of Technology, Czech Republic (to appear during 2007)
7. Erlebach, P.: Automated Pattern-Based Verification for Tree Structures. In: Proc. of the PhD Student Workshop MEMICS'06 (2006)
8. Erlebach, P.: Towards a Systematic Framework for Automatic Pattern-Based Verification of Dynamic Data Structures. In: Proc. of the PhD Student Workshop MEMICS'05 (2005)
9. Lee, O., Yang, H., Yi, K.: Automatic Verification of Pointer Programs Using Grammar-Based Shape Analysis. In: Sagiv, M. (ed.) ESOP 2005. LNCS, vol. 3444, Springer, Heidelberg (2005)
10. Møller, A., Schwartzbach, M.: The Pointer Assertion Logic Engine. In: PLDI'01 (2001)
11. Sagiv, S., Reps, T.W., Wilhelm, R.: Parametric Shape Analysis via 3-Valued Logic. TOPLAS 24(3) (2002)
12. Yavuz-Kahveci, T.: Specification and Automated Verification of Concurrent Software Systems. PhD. thesis, Computer Science Department of University of California, Santa Barbara, CA, USA (2004)

Using Verification Technology to Specify and Detect Malware

Andreas Holzer, Johannes Kinder, and Helmut Veith

Technische Universität München
Fakultät für Informatik
85748 Garching, Germany
{holzera,kinder,veith}@in.tum.de

Abstract. Computer viruses and worms are major threats for our computer infrastructure, and thus, for economy and society at large. Recent work has demonstrated that a model checking based approach to malware detection can capture the semantics of security exploits more accurately than traditional approaches, and consequently achieve higher detection rates. In this approach, malicious behavior is formalized using the expressive specification language CTPL based on classic CTL. This paper gives an overview of our toolchain for malware detection and presents our new system for computer assisted generation of malicious code specifications.

1 Introduction

In the last twenty-five years, model checking has evolved into an industrial-strength framework for the verification of hardware and software. Traditionally, the model checking tool chain assumes that the specifications describe the crucial properties of the system to be analyzed in a positive way, i.e., specifications describe the intended behavior of the system. Our recent approach to malware detection [1] inverts this picture, in that we use specifications to describe malicious behavior. We employ an extension of the temporal logic CTL to specify malicious behavior, and extract a finite state model from the disassembled executable. If the model checker finds out that a specification holds true, then the malware detector reports that the analyzed code is infected. The advantage of our approach over classical malware detection tools is our ability to cover families of malware which use the same attack principle. Our tool is able to detect also previously unknown variants of malware which exhibit behavior similar to that of known malware, but are syntactically different. Classical malware detectors mainly rely on variations of pattern matching using malware signatures from a virus database [2,3]. Thus, they require an update of the virus databases to detect new malware variants.

Malware specifications differ from "standard" software specifications in crucial aspects. Most importantly, a software specification is usually written in the context of the program to be analyzed, i.e., the specification is created with the assistance of the programmer. Variable names, labels, and constant values are often specific to a program; using them in a specification thus requires an understanding of the program. In the typical malware detection scenario, however, we have only little or no knowledge about the program. We usually do not have access to the source code but only to the compiled

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 497–504, 2007.
© Springer-Verlag Berlin Heidelberg 2007

binary or byte code of the software. When the program is indeed malicious, it is very unlikely for the programmer to have created the software in an analysis-friendly way. This scenario creates numerous difficulties specific to malware analysis. First, we need to prepare the program to be analyzed in a suitable manner such that we can extract an abstract model from it. Since the program is in binary form this requires disassembly and, for some files, a decryption mechanism similar to that found in commercial anti-virus tools. Second, the malicious code specification has to be applicable to general programs, that is, it must not contain hard coded variable names. Once the first problem has been successfully addressed, i.e., once a candidate for checking has been disassembled, one has to choose a strategy for extracting the abstract model. This choice heavily influences the nature of specifications that later can be verified against the model. The disassembled binary by itself contains only very low level semantic information about the program. Basically, there are two possible strategies for creating a model from the disassembled program and reasoning about its semantics.

- The first option is to perform extensive preanalysis and to try to extract exact semantics from the assembly code. Specifications for such a model then could be relatively short functional descriptions of malicious behavior. The huge drawback of this method is, however, that the preanalysis requires exact and complete semantics for assembly code. The low level nature of x86 assembly makes an exact functional description infeasible with current technology.
- The second option, which we pursued in our approach, is to implement a coarse abstraction that uses the control flow graph of the program as model and ignores machine state other than the program counter. The resulting model then is a state transition system with one assembly instruction per state. With this approach, specifications become more complex as they need to reflect a lower level of behavior; the need to have abstract variable names and values in specifications is immanent.

Therefore we enriched CTL by variables and quantifiers and obtained a new specification logic CTPL [1]. The advantages of this extension can be easily illustrated by the example specification "there is a register that is first set to zero and later pushed onto the stack", which is on the level of assembly code but abstracts implementation details irrelevant to the malicious behavior. If we try to formalize this specification in CTL, this would result in a large disjunction of the following form:

$$\begin{aligned}&\mathbf{EF}(\texttt{(mov eax,0)} \wedge \mathbf{AF}(\texttt{push eax})) \vee \\ &\mathbf{EF}(\texttt{(mov ebx,0)} \wedge \mathbf{AF}(\texttt{push ebx})) \vee \\ &\mathbf{EF}(\texttt{(mov ecx,0)} \wedge \mathbf{AF}(\texttt{push ecx})) \vee \ldots\end{aligned}$$

CTPL, however, uses predicates rather than atomic propositions to represent assembler instructions, which allows to quantify over an instruction's parameters. In CTPL, we can express the same specification using quantifiers as

$$\exists r\, \mathbf{EF}(\texttt{mov}(r, \texttt{0}) \wedge \mathbf{AF}\, \texttt{push}(r)).$$

Despite the succinct representation CTPL offers, the design of malicious code specifications is a fairly tedious process which involves writing similarly structured formulas

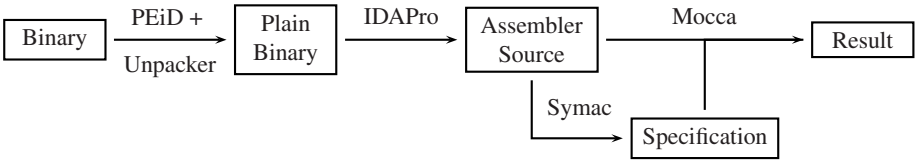


Fig. 1. Malicious Software Detection Process

several times. Therefore, we augmented the specification design phase by implementing *Symac* (**S**pecification s**y**nthesis for **ma**licious **c**ode), a visual editing tool that aids a user in extracting a specification from a representative malware sample. The editor encapsulates many common patterns and provides support for future automated extraction techniques. In this paper, we first give an overview of our malware detection architecture and then proceed to present our new tool for the creation of malicious code specifications in CTPL.

2 Malware Specification and Detection

Figure 1 depicts our complete tool chain for malicious code detection. Our CTPL model checker *Mocca* expects plain text assembly source code as input to construct the internal model representation, so there is some amount of preprocessing necessary when a new executable is to be checked. The majority of malware is *packed*, i.e., encoded using an executable packer, which at runtime decrypts the program into memory. A packed program is practically immune to static analysis and needs to be decrypted before proceeding with the analysis. Thus, as a first step, we have to determine whether the program is packed and which packing mechanism has been used. It is possible to detect packed files by measuring byte entropy or by looking for known patterns generated by common executable packers. For this step, we resort to PEiD [4], a widely used tool for identifying packed files. A number of specialized unpacking programs and libraries are freely available, so knowing which packer was used to protect the program, the corresponding unpacking tool can be chosen to correctly decrypt the executable in the second step. For unknown packers, we can use generic, emulation-based unpacking methods [5].

After unpacking, the resulting plain binary can be passed to a disassembler. We use Datarescue's state-of-the-art disassembler IDAPro [6] for this task, which generates the assembly source code used as input to the Mocca model checker. Mocca creates an abstract model of the executable by parsing the assembly file. During parsing, it performs some simple syntactical substitutions to disambiguate the assembly code (such as replacing `xor eax,eax` with `mov eax,0`). We model assembly code syntactically as Kripke structures, as illustrated in Figure 2. Every instruction is represented by a corresponding predicate, its parameters are treated as constants. Each line of code corresponds to a state in the Kripke structure that is uniquely identified by a so called *location* modeled by the special predicate $\sharp loc$. Transitions in the Kripke structure are added according to the possible control flow of the code: Instructions without successors (e.g. return statements in intraprocedural analysis) are assigned with a self-loop.

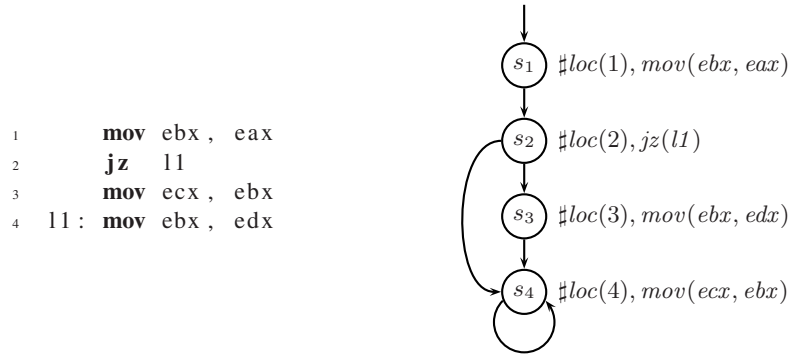


Fig. 2. Example of assembler code and its corresponding Kripke structure

Jumps are connected to their target only, conditional jumps to both possible successors. All other instructions are given a fall-through edge to their successor in the code.

Finally, Mocca checks the model against a malicious code specification in CTPL. Since the specification logic allows quantification, we needed to adapt the bottom-up explicit model checking algorithm for CTL [7] to keep track of possible variable assignments. The introduction of quantifiers causes the CTPL model checking problem to become **PSPACE**-complete [8]. Therefore, the model checker uses several optimizations to reduce the number of procedures checked and to keep the number of tracked variable assignments low. Finally, the model checker reports whether the assembly file satisfies the specification, i.e., whether it is malicious or not.

3 Computer Aided Specification Synthesis

Malware detection in general works by the principle of matching signatures against programs to be scanned. With classical anti-virus tools, nearly every new malware requires an update of the signature database. In our setting, CTPL specifications take the place of malicious code signatures and allow to match whole classes of malware. Due to the broad scope of CTPL specifications, updates are only necessary when a new malware exhibits a novel type of malicious behavior.

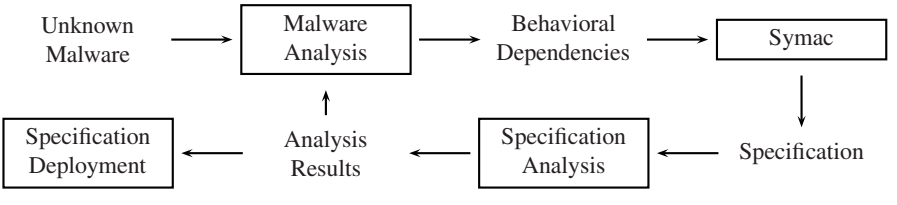


Fig. 3. Tool-supported Specification Generation Process

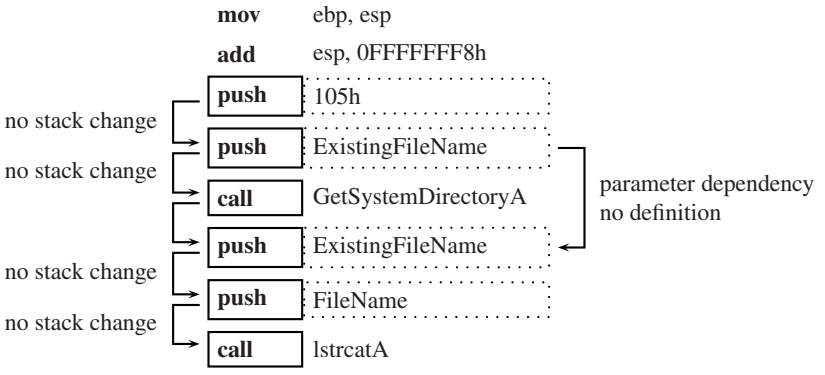


Fig. 4. Code fragment of *Small.aw*, annotated with behavioral dependencies

To create new specifications, we follow the development process shown in Figure 3. The unpacked, disassembled code of new malware is initially loaded into Symac and manually analyzed to locate routines that exhibit characteristic malicious behavior. Once a portion of malicious code is found, we proceed to identify those instructions which are of particular relevance for program behavior. These instructions are typically system or library calls and instructions used for passing data from one call to the other. For example, consider the fragment of assembly code in Figure 4 taken from the Trojan dropper *Small.aw*. It contains two function calls that we identified as characteristic for the malware's behavior. The arguments passed to the calls are written onto the stack by two pairs of push instructions. The string buffer `ExistingFileName` is shared by both function calls. We use *Small.aw* as working example to show how a specification formula is synthesized from malicious assembly code.

The user interactively selects relevant nodes in the control flow graph (selected instructions are enclosed by boxes in Figure 4) and specifies dependencies between them (indicated by arrows). The behavior of the code fragment should be captured by the resulting specification in a general way, so it is important to encode only those dependencies between instructions that are relevant to the behavior. The user can choose between the following different types of dependencies to describe the relevant relationships between nodes:

- **Parameter abstraction:** Parameter abstractions substitute instruction parameters by variables, e.g., to allow the allocation of different registers or memory variables. In our example, the constant `105h` is irrelevant for the description of the malicious behavior and is therefore abstracted away by a variable (indicated by a dotted box).
- **Temporal restriction:** This restriction states that the first instruction has to appear before the second instruction. In our example, temporal restrictions have been added between any two instructions connected by an arrow.
- **No-stack-change restriction:** This restriction states that the first instruction has to appear before the second instruction, and that the stack is not changed by instructions that are executed in-between (in Figure 4, these restrictions ensure the correct parameter setup for the function calls)

$$
\begin{array}{ll}
\Psi_{event}(i) & = \mathbf{EF}\varphi_i \\
\Psi_{temp}(i,j) & = \mathbf{EF}(\varphi_i \wedge \mathbf{EX}(\mathbf{EF}\varphi_j)) \\
\Psi_{stack}(i,j) & = \mathbf{EF}(\varphi_i \wedge \mathbf{EX}(\mathbf{E}[(\forall t.\neg push(t) \wedge \neg pop(t))\mathbf{U}\varphi_j])) \\
\Psi_{def}(i,j,v) & = \mathbf{EF}(\varphi_i \wedge \mathbf{EX}(\mathbf{E}[(\forall v'.\neg mov(v,v') \wedge \neg lea(v,v'))\mathbf{U}\varphi_j])) \\
 & \quad \text{where } v \neq v'
\end{array}
$$

$$
\Phi_{push}(l,t) = \sharp loc(l) \wedge push(t) \qquad \Phi_{call}(l,t) = \sharp loc(l) \wedge call(t)
$$

$$
\begin{array}{lll}
\varphi_1 = \Phi_{push}(l_1, c_1) & \varphi_2 = \Phi_{push}(l_2, dir) & \varphi_3 = \Phi_{call}(l_3, \texttt{GetSystemDirA}) \\
\varphi_4 = \Phi_{push}(l_4, dir) & \varphi_5 = \Phi_{push}(l_5, c_2) & \varphi_6 = \Phi_{call}(l_6, \texttt{lstrcatA})
\end{array}
$$

$$
\exists l_1, l_2, l_3, l_4, l_5, l_6, c_1, c_2, dir.\Psi_{stack}(1,2) \wedge \Psi_{stack}(2,3) \wedge \Psi_{stack}(4,5) \wedge \Psi_{stack}(5,6) \wedge \Psi_{temp}(3,4) \wedge \Psi_{def}(2,4,dir)
$$

Fig. 5. Formula patterns and instantiations corresponding to code fragment in Figure 4

- **No-definition restriction:** This restriction states that the first instruction has to appear before the second instruction, and that there is a parameter of the second instruction that is abstracted by a variable whose value is not changed in-between.
- **Parameter dependency:** Parameter dependency ensures that the mapping of a variable in two instances of parameter abstraction is actually the same. For example, the parameter `ExistingFileName` has to be abstracted by the same variable in both push instructions. The additional no-definition restriction further guarantees that `ExistingFileName` contains the same value.

Symac prohibits cyclic dependencies, allowing a straightforward automatic generation of CTPL formulas using standard graph traversal algorithms. Every element in a finite computation path is represented by a formula $\sharp loc(l) \wedge asmInstr(par_1, \ldots, par_n)$, where the variable l references the location of the element in the Kripke structure, the predicate $asmInstr$ denotes an instruction, and the parameters $par_1, \ldots, par_n$ are either constants or variables. Building upon these basic instruction formulas, Symac generates different types of specification formulas obeying the defined dependencies. Figure 5 shows the patterns Ψ_{stack}, Ψ_{def}, Ψ_{temp}, and Ψ_{event}. The simplest pattern $\Psi_{event}(i)$ just states that some instruction, represented by φ_i, will eventually occur. We handle the restriction to a temporal order between two instruction formulas φ_i and φ_j by instantiating the pattern $\Psi_{temp}(i,j)$. Ψ_{stack} prohibits stack alteration between given instructions. $\Psi_{def}(i,j,v)$ prohibits the redefinition of variable v between two given instructions.

After instantiation of these patterns, the generated formulas are connected by a conjunction. More complex patterns can be achieved by synchronizing individual formulas through the sharing of location variables in multiple location predicates. Every unbound variable is existentially quantified, leading to closed formulas. Finally, the formulas for all single paths are connected by a disjunction. The lower part of Figure 5 shows the instruction formulas for our example and the resulting formula that contains instantiations of the according behavioral patterns.

The final specifications for the Mocca model checker contain a textual and formal description of the corresponding malicious behavior, both generated by Symac. In order

to optimize the model checking process, specifications can also contain *clues*—system calls whose presence in a procedure is implied by the specification formula—that enable Mocca to skip irrelevant procedures from exhaustive analysis. Symac automatically derives these clues from a given CTPL formula [9].

4 Related Work

Commercial anti-virus products still mainly rely on classical detection techniques, such as static string matching. Recently, however, more and more virus scanners have begun using sandboxing and monitoring for detecting suspicious behavior. Szor [2] gives an excellent overview on malware detection and analysis techniques used in the industry today. The Digital Immune System (DIS), introduced by White et al. [10] is a system automating the process of malware analysis and signature generation to some extent. It executes infected binaries in a supervised environment, monitors alteration of the system state and attempts to create a signature from the observed data; if the analysis fails, the system alerts a human specialist. Christodorescu and Jha [11] describe a template based approach to semantic malware detection, particularly focusing on malware obfuscated by a set of common assembly level obfuscations. In follow-up work, they prove completeness of their malware detector with respect to these obfuscations [12].

Dwyer et al. [13] identified common patterns of temporal specifications that can be translated into different temporal logics. Wagner et al. [14] describe a method that automatically derives a model of application behavior in order to detect atypical, suspicious behavior.

5 Conclusion and Future Work

In this paper we presented our malware detection tool chain, including our recent mechanism for specification generation. We implemented the graphical tool Symac, that integrates the process of specification development and enables future automated malware analysis and specification extraction. As a next step, we will investigate to what extent the identification of relevant code and dependencies can be automated. Moreover, we plan to employ automatic analysis techniques such as pattern matching or API extraction [14,15,16]. Further automation of the signature generation process will allow a faster reaction to novel malicious code.

References

1. Kinder, J., Katzenbeisser, S., Schallhart, C., Veith, H.: Detecting malicious code by model checking. In: Julisch, K., Krügel, C. (eds.) DIMVA 2005. LNCS, vol. 3548, pp. 174–187. Springer, Heidelberg (2005)
2. Szor, P.: The Art of Computer Virus Research and Defense. Symantec Press (2005)
3. Christodorescu, M., Jha, S.: Testing malware detectors. In: Avrunin, G.S., Rothermel, G. (eds.) Proceedings of the ACM/SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2004, pp. 34–44. ACM Press, New York (2004)

4. Jibz, Qwerton, snaker, xineohP: PEiD, http://peid.has.it/, Last accessed (May 14, 2007)
5. Christodorescu, M., Kinder, J., Jha, S., Katzenbeisser, S., Veith, H.: Malware normalization. Technical Report 1539, University of Wisconsin, Madison, Wisconsin, USA (2005)
6. DataRescue sa/nv: IDA Pro http://www.datarescue.com/idabase/, Last accessed (May 14, 2007)
7. Clarke, E., Emerson, E.: Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.) Logics of Programs. LNCS, vol. 131, pp. 52–71. Springer, Heidelberg (1982)
8. Kinder, J.: Model checking malicious code. Master's thesis, Technische Universität München (2005)
9. Holzer, A.: Description languages for malicious software. Master's thesis, Technische Universität München (2006)
10. White, S., Swimmer, M., Pring, E., Arnold, W., Chess, D., Morar, J.: Anatomy of a commercial-grade immune system. IBM Research White Paper (1999)
11. Christodorescu, M., Jha, S., Seshia, S., Song, D., Bryant, R.: Semantics-aware malware detection. In: 2005 IEEE Symposium on Security and Privacy (S&P 2005), pp. 32–46. IEEE Computer Society Press, Los Alamitos (2005)
12. Dalla Preda, M., Christodorescu, M., Jha, S., Debray, S.: A semantics-based approach to malware detection. In: Hofmann, M., Felleisen, M. (eds.) Proceedings of the 34th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2007, pp. 377–388. ACM Press, New York (2007)
13. Dwyer, M., Avrunin, G., Corbett, J.: Patterns in property specifications for finite-state verification. In: Proceedings of the 1999 International Conference on Software Engineering (ICSE'99), pp. 411–420. ACM Press, New York (1999)
14. Wagner, D., Dean, D.: Intrusion detection via static analysis. In: 2001 IEEE Symposium on Security and Privacy (S&P 2001), may 2001, pp. 156–169. IEEE Computer Society Press, Los Alamitos (2001)
15. Liu, C., Ye, E., Richardson, D.J.: Software library usage pattern extraction using a software model checker. In: 21st IEEE/ACM International Conference on Automated Software Engineering (ASE 2006), pp. 301–304. IEEE Computer Society Press, Los Alamitos (2006)
16. Ammons, G., Bodík, R., Larus, J.: Mining specifications. In: Symposium on Principles of Programming Languages, pp. 4–16. ACM Press, New York (2002)

A Compositional Approach for Equivalence Checking of Sequential Circuits with Unknown Reset State and Overlapping Partitions

Gabriel P. Bischoff[1], Karl S. Brace[1], and Gianpiero Cabodi[2]

[1] Massachusetts Microprocessor Design Center
Intel Massachusetts Inc, Hudson, MA, USA
{gabriel.bischoff,karl.brace}@intel.com
[2] Dip. di Automatica e Informatica, Politecnico di Torino, Turin, Italy
gianpiero.cabodi@polito.it
http://fmgroup.polito.it/cabodi

Abstract. This paper describes novel contributions to the problem of sequential equivalence checking. We address industrial setups, where the design of VLSI chips typically requires checking the equivalence of an RTL model (the specification) and a gate level optimized circuit (the implementation). Due to the size of the overall problem, compositionality is required. The circuit must be resetable, but the reset state is not yet known when equivalence checking is performed. In this paper we discuss the conditions under which decomposed proofs of equivalence are able to infer the equivalence of the full design. Our main contributions with respect to the state of the art in this field are: (1) discussing compositionality given a 3-valued initialization scheme, (2) accepting decompositions with overlapping partitions.

1 Introduction

Formal equivalence checking is the process of proving that two designs are "functionally equivalent". Most industrial verification platforms presently support combinational equivalence checking, whereas no sequential optimizations are allowed. Two designs (e.g. an RTL specification and a synthesized gate level implementation) are proved (input-output) equivalent if they have the same state encodings, and their corresponding (i.e. mapped) combinational sub-circuits are functionally equivalent. However, due to the increasing pressures to synthesize smaller and faster designs, more aggressive sequential optimizations are performed. The process of verifying the equivalence of two circuits becomes a sequential issue. In principle, a sequential equivalence proof is similar to a model checking problem [1], with two additional complications w.r.t. standard property checking frameworks:

- The initial (reset) states of the two circuits are generally not known at the time the designs are checked, and
- It is often necessary to exploit some knowledge of the problem structure and/or synthesis/optimization methodology.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 505–514, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Furthermore, due to the inherent size and complexity of the checked designs, equivalence proofs are performed upon decomposed circuits. A computational framework for sequential equivalence checking was introduced in [2]. An implementation exploiting BDD and ATPG based techniques was then proposed in [3,4]. More recently, an industrial implementation, fully relying on SAT engines, was described in [5,6], and a simpler approach, mixing combinational and sequential equivalence, was described in [7]

Given two Finite State Machines (FSMs) under check (and/or their product machine), the above mentioned works define

- The conditions under which an FSM can be synchronized and/or initialized (by a given input vector) to a known state, starting from an arbitrary (unknown) power-on state, and
- Equivalent states of the two FSMs, i.e., those states that guarantee undistinguishable present and future I/O behavior for the FSMs, and
- Conditions for alignability, safe replaceability, and equivalence, taking into account the need to initialize the two FSMs, and
- Compositionality conditions for equivalence proofs.

The best result attained is the ability to decompose a sequential verification problem onto a set of non-overlapping partitions. In [6], the authors introduce a "stable decomposition". They also support additional environment assumptions ("verification properties"), that enable them to refine the abstract models generated by decompositions (and to reduce false mismatches).

As an alternative (and complement) to their work, we propose here to make use of 3-valued logic initialization and to determine equivalence by equivalent state pairs, and to further filter out false mismatches by allowing overlapping partitions.[1]

2 Background and Related Works

The sequential systems we address are usually modeled as Finite State Machines (FSMs).

A Finite State Machine is defined as a tuple M = (S, Σ, Γ, δ, λ), where: S is the set of states, Σ is a finite input alphabet, Γ is a finite output alphabet, $\delta : S \times \Sigma \rightarrow S$ is the state transition function, $\lambda : S \times \Sigma \rightarrow \Gamma$ is the output function.

A major difference of sequential equivalence checking of FSMs, w.r.t. model (property) checking, is that the former is usually operated without the knowledge of an initial state, which is typically available in model checking problems.

We briefly summarize some pertinent terminology from the literature. An *equivalent state* is a state in two circuits such that starting from that state, no

[1] Although the word "partition" sounds like it implies non-overlap, following convention we use the term for both overlapping or non-overlapping logic cones.

binary input sequence will cause any corresponding circuit outputs to differ. A circuit is *weakly synchonizable* if there exists a binary input sequence that drives it to a set of equivalent states from any arbitrary binary state. A circuit is *synchronizable* if there exists a binary input sequence that drives it to a particular state from any arbitrary binary state. A circuit is *3-valued X-initializable* (or simply *initializable*) if there exists a binary sequence that will drive the circuit to a known state from the all-X state using a 3-valued simulator. Two circuits are alignable if, for every state set in one, there exists an equivalent state in the other. A circuit C_0 is *3-value safe replaceable* by C_1 if, starting from the all-X state, no 3-valued input sequence causes any output of C_0 to be binary while the corresponding output of C_1 differs from it.

Within this framework, state of the art theory [8] either resorts to the concepts of alignability for (weakly) synchronizable circuits, or 3-value safe replaceability for (3-valued) initializable circuits. Initializability is a stronger condition: an initializing sequence is also a (weak) synchronizing sequence, whereas the reverse implication does not hold.

Sequential equivalence checking of industrial designs has been addressed in a number of other papers. All rely on the concept of *compositionality*, which is the property that if the design is divided into smaller components, and each of those components is found to be equivalent, then the design as a whole can be declared equivalent. Compositionality depends upon the method used to check the components and restrictions on the nature of the components and/or the overall design.

Many verification methods, as described here, require knowledge of the global reset state, which we do not require. Mony et al. [9] developed a method they call Transformation Based Verification, which locates structural similarities to expose equivalences, and applies a variety of partition verification methods to those that require more effort. Huang et al. [3,4] also exploit structural similarities between the specification and implementation to simplify the verification task. They also make use of *safe replaceability*, which is safely decompositional, with non-overlapping partitions. Stoffel [10] describes a method based on a non-canonical (overapproximation) method of state machine analysis.

Our work is more similar to that of Khasidashvili [6] and of Moon [7] in that we require no global reset state. Khasidashvili [6] shows that a design composed of non-overlapping partitions is compositional with respect to alignability. Moon's method [7] is even more similar to ours in that sequential equivalence is defined in terms of a non-empty set of Equivalent State Pairs (ESPs). We used this notion for verification of a microprocessor design [11] but it was previously described by Coudert et al. [12]. Moon shows that if the partitions are non-overlapping (a *well partitioned design*) ESP-based verification is compositional. Our contribution in this paper is also an extension of [3,4], as we define the restrictions on initializability and 3-valued safe replaceability necessary to extend ESP compositionality to designs containing overlapping partitions.

3 Compositionality with Overlapping Partitions

3.1 The Verification Framework

In our verification scheme, the global circuit is composed of a *Spec* (specification) circuit, an *Imp* (implementation) circuit, and *Imp properties*. *Imp properties* are driven by *Imp* signals. In addition, there are *mapping properties* that are driven by both *Spec* and *Imp* signals. A *partition* is a cone of logic, including both *Spec* and *Imp* logic and usually property logic as well. A partition output is generally property logic driven by *Spec* and/or *Imp* logic. For example, a *mapping property* partition is an equivalence gate driven by mapped *Spec* and *Imp* signals. A partition has inputs that can be any cut in the fanin logic. The partition also includes the mapping properties to relate the *Spec* and *Imp* signals to each other. *Imp* properties are verified but are not used for model refinement.

Spec properties that are used to constrain the verification are called *assumptions*. These serve to reduce false mismatches, and include mapping properties on partition inputs (mapped *Spec* and *Imp* signals are equal) and logical properties on *Spec* inputs and intermediates (such as mutual exclusion). Such properties can be temporal/sequential as well. However, for this research we consider only mapping properties. The more complex assumption properties usually serve to define a *don't-care* that allows the *Imp* and *Spec* to differ when the don't-care holds. However, because the don't-care information is often enforced by nearby fanin logic, a false mismatch can typically be avoided by *expanding* the partition through its immediate fanin so that it includes the logic that enforces the don't-care. Such expansion causes extensive overlap between the fanout and fanin partitions, and also between all fanout partitions that include that don't-care-generating logic. This is a major source of overlap between partitions in our verification framework.

In our framework we assume the following:

- There exists a defined initialization sequence Sr that, when applied to mapped primary inputs in a 3-state simulator, will remove all power-on X's from state elements in both the *Spec* and *Imp*, including their respective properties, and
- Partitions might overlap, that is, separate partitions might share *Spec*, *Imp*, or property logic, and
- A partition might extend through state elements, and include synchronous feedback and mapping properties of inputs, and
- Every logic element in the *Imp* is part of some partition or another, and
- Each partition, analyzed separately, satisfies the property that there exists a non-empty *"equivalent state pairs (ESP) set"* such that, if the partition gets into any of these states, all input sequences that satisfy all of its assumption properties will satisfy it, and
- Properties that depend only on primary inputs need not be verified.

3.2 Equivalent State Pairs and Compositionality

Due to the size of designs under verification, equivalence checks are only achievable on partitions and/or decomposed circuits. This means that compositionality

of equivalence proofs is a key issue when defining the overal verification methodology. As a first remark, recall that there are problems with the standard approaches to defining sequential equivalence in the absence of reset (or initial) states. Some notions, like alignability [8], are not by nature compositional, as equivalence of parts does not imply equivalence of the top-level designs. The extension described in [6] is compositional, with the addition of manual properties. Other notions of equivalence such as safe replacement [4] are compositional, but, as in [6], no overlapping partitions are considered. The ESP solution [7] (Two circuits are sequentially equivalent precisely when they have a nonempty set of equivalent state pairs) is compositional, again provided that the partitions do not overlap, in particular that they do not share any state elements.

The intuition under all compositionality approaches (with no overlap) is this: If each partition has a non empty set of equivalent states, the Cartesian product of all those states is non void by definition, so the overall circuit has a non empty set of equivalent state pairs.

3.3 Compositionality and Synchronization

Weak synchronization is not compositional because the behavior considered for individual partitions under verification misses legal behavior of the overall circuit. This is taken into account in [6] by the addition of additional constraints (verification properties). Yet compositionality does not allow overlapping. In a nutshell, the problem can be synthesized as follows:

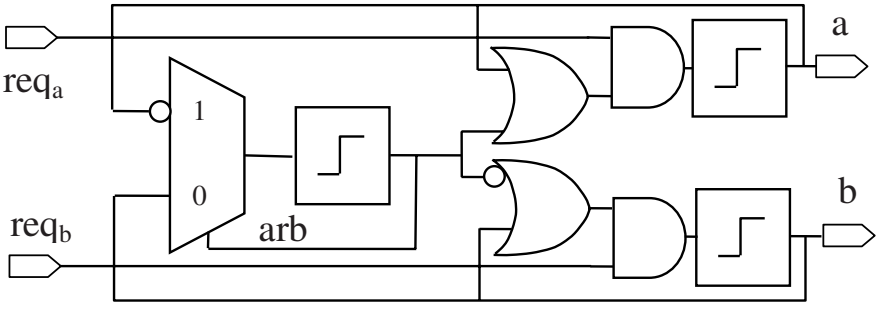


Fig. 1. Arbiter with two overlapping partitions

Two overlapping partitions M_0 and M_1 (with shared state elements) can (individually) have equivalent state pairs, yet the intersection between the state pair set can be void, so the global Machine M_0, M_1 has no equivalent state pairs. This can happen even in the case of a weakly synchronizable M_0, M_1 circuit. For example, consider the arbiter circuit *Spec* in Figure 1. The clock has been omitted from the figure for clarity, but is an input in the actual design. The arbiter works as follows: The output signals a and b enable a pair of resources in turn. The signals req_a and req_b request permission for a and b, respectively.

The signal *arb* keeps track of which resource received permission last. Suppose that the *Imp* for this *Spec* is simply logical 0 (which is clearly an error), with mapping properties specifying $a = 0$ and $b = 0$. This circuit is fully resetable from any arbitrary binary power-on state by setting (req_a,req_b) to 10, 00, 01, 00 on 4 consecutive clock cycles, leaving arb=1 (a's turn). It can be decomposed into two overlapping partitions. As illustrated in Figure 2, the two partitions share some logic including the state element *arb*, which we have labeled *arb*′ in the second partition to indicate that it is analyzed separately from *arb*. The $a = 0$ partition includes the input mapping $b = 0$ and we find that $a = 0$ and $arb = 0$ is a equivalent state. Intuitively, $arb = 0$ means it is b's turn, and if it is b's turn, and b is always 0, a will always be 0. Similarly, the $b = 0$ partition has an equivalent state $b = 0$ and $arb = 1$ and input property $a = 0$, which means that if it is a's turn and a is always 0, b will always be 0. The mapping of a and b to 0 in the *Imp* is clearly wrong, but because each partition has a non-empty ESP, if we assumed compositionality we would incorrectly declare the *Imp* and mapping to be verified!

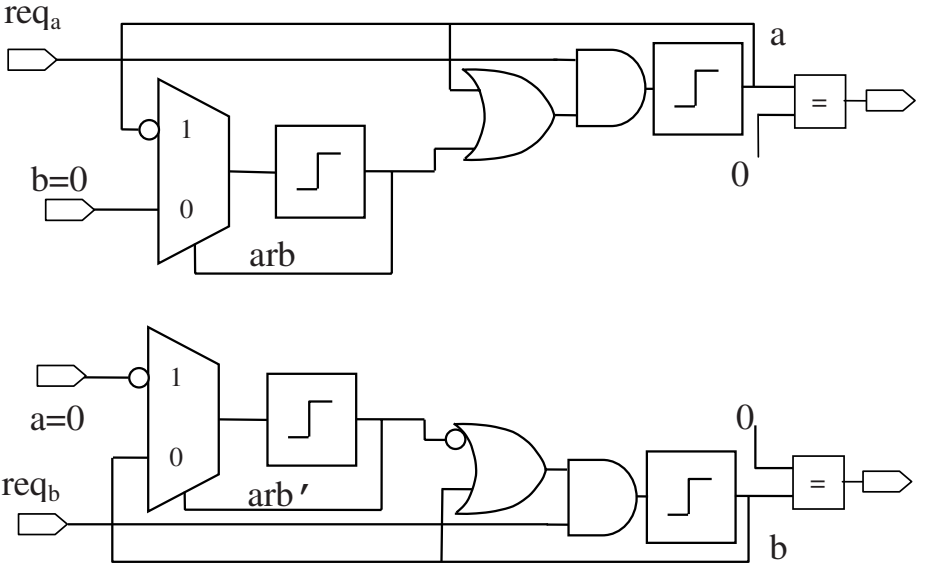


Fig. 2. Duplicated logic in overlapping partitions

This example illustrates that ESP-based verification of a weak synchronizable circuit is not compositional, because the equivalent state pairs for the two partitions have a null intersection, in that one requires $arb = 0$, the other $arb = 1$. The only way to guarantee that the intersection of ESPs is non null for a weak synchronizable circuit is to determine individual ESPs on non-overlapping partitions.

3.4 Compositionality and (Three-Valued) Equivalence

3-valued self replaceability is a stronger condition than weak-synchronization, i.e. if a machine is resetable, it is also weakly synchronizable, but the reverse is not true.

Our approach is similar to 3-valued self replaceability [4], as we operate on a ternary (3-valued) model, and we assume the existence of an initializing sequence, able to bring the circuit from the unknown to a fully binary state (in the 3-valued space). Our work differs from [4], as our notion of equivalence is based on ESPs: equivalence relies on the existance of a non void set of equivalent binary states, which is computationally easier to determine.

Let us take a circuit M under verification, and decompose it onto a set of partitions such that the partitions overlap (share state elements). Without loss of generality, let us consider the case of two partitions M_0 and M_1:

$$M = (S, \Sigma, \Gamma, \delta, \lambda)$$
$$M_0 = (S_0, \Sigma_0, \Gamma_0, \delta_0, \lambda_0)$$
$$M_1 = (S_1, \Sigma_1, \Gamma_1, \delta_1, \lambda_1)$$

The M_0 and M_1 partitions are overlapping if S_0 and S_1 share some (common) state variables. Let x_0 (x_1) be the set of state variables of M_0 (M_1). The condition for overlapping is that $x_{01} = x_0 \cap x_1 \neq \emptyset$.

Consider a machine $\hat{M}$ that has non-overlapping partitions, generated from M as follows. First copy M_0 into $\hat{M}$ to form $\hat{M}_0$. Then copy M_1 into $\hat{M}$ so that $\hat{M}_1$ shares only primary inputs and perhaps the output of $\hat{M}_0$ (if M_1 has M_0's output as an input). If M_0 and M_1 overlap, that logic and any shared state elements appear twice in $\hat{M}$. Note that if M_1 is in M_0's fanout, but M_1 had been expanded to include M_0's logic, then M_0's logic will be duplicated in $\hat{M}$, and $\hat{M}_0$'s output will be a dangle in $\hat{M}$. This is a common source of overlap in our framework, but such overlap and dangles do not complicate our analysis. State elements that are part of both partitions will thus be duplicated in $\hat{M}$. Let $\hat{x}_{01}$ be the set of variables in $\hat{M}$ that correspond to the overlapping variables x_{01} in M_0 and M_1.

$\hat{M}$ is thus decomposed in non-overlapping partitions. The state space of machine $\hat{M}$ is obviously larger than M's state space, due to the higher number of state variables.

Furthermore, the behavior of $\hat{M}$ includes that of M. More formally, whenever $\hat{x}_{01}$ variables are constrained to hold the same values as their corresponding x_{01} variables, the two machines are equivalent. Let $\hat{S}$ be the state space of $\hat{M}$, and let $\hat{S}_{eq} \subseteq \hat{S}$ denotate the subspace characterized by equal values for x_{01} and $\hat{x}_{01}$ variables.

$$\forall \sigma_i \in x_{01}, \sigma_i = \hat{\sigma}_i$$

The state transition graph (STG) of M is isomorphic to the subset of the $\hat{M}$ STG included in $\hat{S}_{eq}$.

So $\hat{M}$ is not equivalent to M, but equivalence is guaranteed whenever $\hat{M}$ works in the above mentioned $\hat{S}_{eq}$ sub-space.

Let us now point out the following **observations**:

- If M is (weakly) synchronizable, this does not imply that $\hat{M}$ is (weakly) synchronizable, due to the extra behavior of $\hat{M}$ for states where, for some i, $\sigma_i \neq \sigma_i'$
- However, M is 3-valued initializable iff $\hat{M}$ is 3-valued initializable, as the unknown state is a single state in the ternary domain, and it is within S_{eq}. So M and $\hat{M}$ are equivalent when starting from the unknown state.

Intuitively, even if M can be reset from an arbitrary binary state, it is still possible that $\hat{M}$ will initialize differently. For example, suppose M includes an XOR of two complementary signals that were derived from a single state element and its complement. In M this XOR gate will be driven to 1 for any initial value in that state element, but in $\hat{M}$ that state element will be duplicated, and the XOR gate can be driven to 0 if the two copies of that state element have different initial values. This logical difference could affect the final reset state of the machine. But if M is X-resetable, it can reset even if an initial X reconverges, so such binary consistency in the initial state is not necessary, thus M and $\hat{M}$ reset to equivalent states for any initial values of the duplicated states.

Let us now formulate the theorem stating our main contribution.

Theorem 1. *Suppose that M is divided into overlapping partitions, and equivalence holds in each partition M_i, under the assumption that the others (and their respective mapped outputs) are equivalent, and M is 3-valued initializable. Then equivalence holds on M, i.e. overlapping partitions are compositional.*

Sketch of proof. Transform M into $\hat{M}$ with non overlapping partitions, by means of the previously described transformation. If M was 3-valued initializable, then also $\hat{M}$ is. For each partition M_i, let us consider a power-on state E_i where the state elements hold an arbitrary value chosen within the ESP_i, that was determined for that partition.

- Such a state exists for each partition because each partition has been proved equivalent, but
- The states might be outside S_{eq}, because we might arbitrarily choose a state with different values for x_{01} and $\hat{x}_{01}$.

Now let us apply the initializing sequence to $\hat{M}$ starting from $E = \bigcap_i E_i$ (cartesian product of equivalent power-on states). The sequence will bring $\hat{M}$ to the reset state I computed on the ternary model, which is the same for M and $\hat{M}$ (modulo projection onto the x_{01} variables, see *observations*).

As the power-on state E is an equivalent state for all partitions, all partition input properties hold at each step during initialization, thus the reset state I itself is an equivalence state for all partitions in $\hat{M}$ (the projection onto each partition is an ESP). Also, $\hat{M}$ initializes all copies of the duplicated state elements with equal values, thus the reset state is in $\hat{S}_eq$, and it is included in the ESPs of all partitions. This proves that the intersection of the separate partition ESPs is not void and thus equivalence is compositional.

Intuitively, the individual ESP verifications showed that an equivalence state exists for each partition. We can create a non-overlapping machine $\hat{M}$ that initializes from an arbitrary binary state to the same state as M (with duplicate state elements having the same value as the originals) only because M is X-initializable. Thus we know that if we superimpose the ESPs from the component verifications onto $\hat{M}$ and apply the reset sequence, all mapping properties will hold at every step. Because a full reset *can* occur without violating any properties (if we are lucky and power-on in a state that honors all component ESPs), we know the reset state itself is in the ESP (otherwise a component verification would have failed), no matter how it is reached.

4 Conclusions and Future Work

We have shown that the sequential verification of equivalent state pairs of synchronizable circuits is not compositional when verification slices are overlapping. We have also shown that the sequential verification of equivalent state pairs of initializable circuits *is* compositional when verification slices are overlapping. This result is relevant to the verification of equivalent state pairs in the presence of constraints (*Spec properties*) because the verification of input constraints together with the verification of equivalent state pairs implies overlapping partitions. We will address in future work compositionality of equivalent state pairs with general constraints and overlapping partitions.

References

1. Grumberg, O., Clarke, E., Peled, D. (eds.): Model Checking. MIT Press, Redmond, Washington (2000)
2. Pixley, C.: A Theory and Implementation of Sequential Hardware Equivalence. IEEE Trans. on Computer-Aided Design 11(12), 1469–1478 (1992)
3. Huang, S.Y., Cheng, K.T., Chen, K.C., Brewer, F., Huang, C.Y.: AQUILA: An equivalence checking system for large sequential designs. IEEE Trans. on Computers (2000)
4. Huang, S.Y., Cheng, K.T., Chen, K.C.: Verifying sequential equivalence using ATPG techniques. ACM Transactions on Design Automation of Electronic Systems (2001)
5. Khasidashvili, Z., Hanna, Z.: SAT-based methods for sequential hardware equivalence verification without synchronization. In: Electronic Notes in Theoretical Computer Science, Elsevier, North-Holland, Amsterdam (2003)
6. Khasidashvili, Z., Skaba, M., Kaiss, D., Hanna, Z.: Theoretical framework for compositional sequential hardware equivalence verification in presence of design constraints. In: Proc. Int'l Conf. on Computer-Aided Design, San Jose, California, pp. 58–65 (November 2004)
7. Moon, I., Bjesse, P., Pixley, C.: A Compositional Approach to the Combination of Combinational and Sequential Equivalence Checking of Circuits without Known Reset States. In: Proc. Design Automation & Test in Europe Conf, Nice, France (April 2007)

8. Pixley, C.: A Computational Theory and Implementation of Sequential Hardware Equivalence. AMS/DIMACS Series in Discrete Mathematics and Theoretical Computer science 3, 293–320 (1991)
9. Mony, H., Baumbartnet, J., Paruthi, V., Kanzelman, R.: Exploiting suspected redundancy without proving it. In: Proc. IEEE DAC 2005, San Diego, CA, pp. 463–466 (June 2005)
10. Stoffel, D., Wedler, M., Warkentin, P., Kunz, W.: Structural FSM traversal. IEEE Transactions on Computer-Aided Design (May 2004)
11. Bischoff, G.P., Brace, K.S., Jain, S., Razdan, R.: Formal Implementation Verification of the Bus Interface Unit for the Alpha 21264 Microprocessor. In: Proc. Int'l Conf. on Computer Design, Austin, Texas (October 1997)
12. Coudert, O., Berthet, C., Madre, J.C.: Verification of Sequential Machines Based on Symbolic Execution. In: Sifakis, J. (ed.) Automatic Verification Methods for Finite State Systems. LNCS, vol. 407, pp. 365–373. Springer, Heidelberg (1990)

Efficient Model Checking of Applications with Input/Output

Cyrille Artho[1], Boris Zweimüller[2], Armin Biere[3], Etsuya Shibayama[1], and Shinichi Honiden[4]

[1] Research Center for Information Security (RCIS), National Inst. of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
[2] Computer Systems Institute, ETH Zürich, Switzerland
[3] Johannes Kepler University, Linz, Austria
[4] National Institute of Informatics, Tokyo, Japan

Abstract. Most non-trivial applications use some form of input/output (I/O), such as network communication. When model checking such an application, a simple state space exploration scheme is not applicable, as the process being model checked would replay I/O operations when revisiting a given state. Thus software model checking needs to encapsulate such operations in a caching layer that is capable of hiding redundant executions of I/O operations from the environment.

Keywords: Software model checking, network communication, software testing.

1 Introduction

Model checking explores the entire behavior of a system under test (SUT) by investigating each reachable system state [5] for different thread schedules. Recently, model checking has been applied directly to software [2,4,6,7,13]. However, conventional software model checking techniques are not applicable to networked programs. The problem is that state space exploration involves backtracking. After backtracking, the model checker will execute certain parts of the program (and thus certain input/output operations) again. However, external processes, which are not under the control of the model checking engine, cannot be kept in synchronization with backtracking, causing direct communication between the SUT and external processes to fail.

Our work proposes a solution to this problem. It covers all input/output (I/O) operations on streams and is applicable as long as I/O operations of the SUT always produce the same data stream, regardless of the non-determinism of the schedule.

This paper is organized as follows: An intuition for our algorithm is given in Section 2, while Section 3 formalizes our algorithm. Experiments are given in Section 4. Section 5 describes related work. Future work is outlined in Section 6, which concludes this paper.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 515–522, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Intuition of the Caching Algorithm

Model checking of a multi-threaded program analyzes all non-deterministic decisions in a program. Non-determinism includes all possible interleavings between threads that can be generated by the thread scheduler. Alternative schedules are explored by storing the current program state and executing copies of said program state under different schedules. When model checking a SUT that is part of a distributed system using multiple processes, external processes are not backtracked during model checking. Thus, two problems arise:

1. The SUT will re-send data after backtracking. This will interfere with the correct functionality of an external process.
2. After backtracking, the SUT will expect external input again. However, an external process does not re-send previously transmitted data.

One possible solution to this problem is to lift the power of a model checker from process level to operating system (OS) level. This way, any I/O operation is under control of the model checker [10]. However, this approach suffers from scalability problems, as the combination of multiple processes yields a very large state space.

Similar scalability problems arise if one transforms several processes into a single process by a technique called *centralization* [11]. With a model for TCP/IP, networked applications can be model checked, but the approach does not scale to large systems [1,3].

Our approach differs in that it only executes a single process inside the model checker, and runs all the other applications externally. Inter-process communication is supported by intercepting any network traffic in a special cache layer. This cache layer represents the state of communication between the SUT and external processes at different points in time. After backtracking to an earlier program state, data previously received by the SUT is replayed by the cache when requested again. Data previously sent by the SUT is not sent again over the network; instead, it is compared to the data contained in the cache. The underlying assumption is that communication between processes has to be independent of the thread schedule. Therefore, the order in which I/O operations occur must be consistent for all possible thread interleavings. If this were not the case, behavior of the communication resource would be undefined. Whenever communication proceeds beyond previously cached information, the new data is both physically transmitted over the network and also added to the cache. The only exception to this is closing a connection. The cache simulates the effect of closing communication but allows connections to remain physically open for subsequent backtracking.

For this approach to work, communication with the environment must be independent of thread scheduling. Therefore, the order in which I/O operations occur and data is sent, must be consistent for all possible thread interleavings. If this were not the case, behavior of the communication resource would be undefined, as all communication is assumed to be deterministic.

3 Formalization of Stream I/O with Rollback

The following definitions assume a semantics for variables as in computer programs, allowing for updates of variables and functions using assignment operator :=.

3.1 Stream Abstraction

Programs operate on a set S of data streams (which correspond to streams or sockets in a given programming language). A communication trace t is a finite sequence of data: $t = \langle t_0, \ldots, t_i \rangle$. Without loss of generality, we assume a uniform size for all data values used. Let $|t|$ denote the length of a trace and T the set of all traces. A data stream $s\,\langle t, st \rangle$ consists of a communication trace t and a stream state st. A stream state $st\,\langle c, p \rangle$ consists of a connection state c where $c \in \{open, closed\}$, and the current position p in the associated communication trace. Function $\mathrm{t}(s)$ returns the corresponding trace of a data stream; $c(s)$ and $p(s)$ return the connection state of a stream state and its position, respectively. The current stream state for a given stream s is returned by function state : $s \rightarrow st$.

A milestone m consists of the full program state, including current stream states. Stored stream states are modeled by function mstate. This function returns a function containing stored stream states, mstate : $m \rightarrow$ state. Let "linear mode" denote execution when no milestone is active, and "milestone mode" an execution trace during which at least one milestone has been created (and not yet removed). In milestone mode, the model checker maintains a set of all milestones M.

3.2 Execution Semantics

Model checking of a program using I/O is performed as follows: In linear mode, all operations are directly executed, using the functionality provided by the standard library. The result of the library function, using the correct set of parameters, will be denoted by `lib`(...). In milestone mode, all subsequent changes to communication traces are recorded in T. Communication traces in T are recorded globally, outside each milestone.

A communication trace of a given stream, $t(s)$, is *consistent* w.r.t. a previously seen communication trace $t \in T$ iff, for the current schedule, the same trace is encountered up to position $p(s)$ of stream s. In our approach, all communication traces have to be consistent with the first seen communication trace. During any non-deterministic run of a program, there has to be one unique communication trace t' such that for all thread schedules, $t(s) = t'$.

All data sent over a stream has to be equal across all possible program schedules, as the schedule should not change application behavior. Actions extending the system state beyond a previously cached state extend cache information with new data. In order to denote this, position p reflects the current position of the cached trace, which is increased when the current trace exceeds a previously cached one, as defined below.

Creation of a milestone m requires the model checker to record the current state of all streams in m, i.e., mstate(m) := *state*. In milestone mode, *execution* behaves as follows:

- Reading data: `read`(s) returns $\begin{cases} t_{p(s)}(s) \text{ if } p(s) \leq |t(s)| \\ \texttt{lib}(\ldots) \text{ otherwise} \end{cases}$
  This operation also sets $t_{p(s)}$ to the value returned and increments $p(s)$ after that.
- Writing data: `write`(s, d) $\begin{cases} \text{checks if } t_{p(s)}(s) = d \text{ if } p(s) \leq |t(s)| \\ \text{calls } \texttt{lib}(\ldots) \text{ and sets } t_{p(s)} := d \text{ otherwise} \end{cases}$
  If $t_{p(s)} \neq d$, the program trace is inconsistent with a previously checked schedule, and model checking is aborted. Otherwise, $p(s)$ is incremented after access to $t_{p(s)}$.
- Opening a stream: `open`(s) $\begin{cases} \text{returns an error if } c(s) = \textit{open} \\ \text{sets } c(s) := \textit{open} \text{ otherwise} \end{cases}$
- Closing a stream: `close`(s) $\begin{cases} \text{returns an error if } c(s) = \textit{closed} \\ \text{sets } c(s) := \textit{closed} \text{ otherwise} \end{cases}$
  The error codes returned for `open` and `close` correspond to the ones returned by `lib` in the same situation.

A *rollback* operation affects the state of each stream in m, restoring them to their previous value: state := mstate(m). The cached communication trace t of each data stream is not reverted. If previously recorded parts of a communication trace are re-sent by the model checker, they are only accepted if they match the given history. Changes in t reflect the fact that the current exploration sent more data to the network than in previously seen subsets of the state space.

When a milestone is *removed* from M, its associated state information is discarded. Communication traces are stored and mapped as long as milestones containing them exist. A stream is physically closed when the last milestone containing it is removed and $c(s) = \textit{closed}$.

3.3 Limitations of Replay-Based Approaches

Any program whose communication fulfills the criteria defined above can be model checked successfully using our approach. However, there are classes of programs that are normally considered to be valid, for which our criteria are too strict. This includes software that logs events to a file or network connection. For this discussion it is assumed that logging occurs by using methods `open`, `write`, and `close`. Assume further that actions of each thread can be interleaved with actions of other threads, which include logging.

If log entries of individual threads depend on thread-local data, they are independent of each other. In such a case, different correct interleavings of log entries can occur without violating program correctness. If log data is sent over a single shared communication channel, occurrence of different message interleavings violates the criterion saying that written data at a specific position must be equal for all thread interleavings. Such programs can therefore not be model checked

with our approach, unless some messages were treated specially, e.g. by ignoring the order in which they appear in the trace.

Note that this limitation only applies if several threads share the same connection. If each thread has its own connection, then the order in which connections are used may be affected by the schedule, but as long as the content of each communication trace does not vary across schedules, our consistency criterion is fulfilled.

On a more general level, applications where communication depends on the global application state are not applicable to our approach. A chat server that responds to requests by sending the number of currently connected clients is an example for this. Communication (the server response) depends on the total number of clients connected. Assume a chat *client* is run inside the model checker, which has two threads connecting to the server. When replaying a partial communication trace of one thread, this communication trace may not match with a previously seen trace, because the number of clients currently connected varies depending on the thread schedule. Similar problems appear if communication content depends on the state of other processes. Such cases can only be model checked by using application centralization [1,3].

3.4 Limitations of Our Implementation

Our approach is strictly stream-based. Our initial implementation does not properly distinguish between communication channels used by different threads, and therefore does not work on more complex applications. While we have successfully run our tool on a simple web server, where each request and response consists of a single, atomic message, the implementation fails for more complex protocols. Work is in progress to address this problem.

4 Experiments

It is not obvious how communicating applications can or should be tested. Clearly, it is necessary to have at least two communicating applications: the SUT running inside a model checker, and the remote application running independently. Even though applications can be truly distributed, running on different hosts, it is necessary to execute them on the same host for expedient automation of such a test. In order to allow for easier automation, both the internal and "remote" application are launched by our model checker, JNuke.

Both the SUT and the remote application have to be synchronized for initiating a test. If this was not the case, it could happen that a client attempts to contact a server that is not ready yet. Indeed, it is not trivial to avoid such a scenario, as the state of the external application cannot be supervised or influenced by the model checker. There are two ways to prevent premature client startup:

1. Extra control code could be added to the client, ensuring that the server is ready. For instance, the client could retry a communication attempt in the event of failure.

2. Starting the client is delayed after starting the server. This allows the server to initialize itself and be ready for communication.

The second approach is less reliable, but more practical, as it does not require modification of the SUT. Experiments with this approach worked quite reliably under different settings. Reliability could be further improved by using operating system utilities to supervise system calls. Such tools include `trace`, `strace`, and `truss` [8]. Unfortunately, it is not possible to access them in a uniform manner from inside an application on different platforms.

Initial experiments tested the performance of our I/O layer with no model checking enabled (linear execution). These tests were run on two Pentium 4 computers with a clock frequency of 1.3 GHz and 512 MB of RAM. Table 1 shows that our implementation is not yet quite as fast as the one in Sun's VM but delivers a comparable performance for sending larger bursts of data (more than 1 KB per call). For experiments, 100 MB were transmitted. User time (the time spent while executing bytecode and native methods in the VM) and total time (real time) were measured. The difference indicates the time needed to execute the system calls of the operating system.

Table 1. Performance of network I/O when transmitting 100 MB

Bytes per `read`/`write` operation	Sun's VM [s]		JNuke VM [s]	
	User	Total	User	Total
100	2.3	8.7	35.1	58.1
1,000	0.5	8.7	11.7	19.2
10,000	0.4	8.7	8.9	13.9

We have used the same approach to test application behavior when the target application runs in model checking mode, communicating with external applications. Implementation problems with disambiguating streams across multiple threads have lead to failures for sample applications that involve more than a single message per channel. Work in progress addresses these problems.

5 Related Work

Software model checkers [2,4,6,7,13] store the full program state (or differences to a previously stored state) for backtracking. They are typically implemented as explicit-state model checkers. Milestone creation and rollback operations occur many times during state space exploration. This causes operations to be executed several times when a set of schedules is explored. Such exploration does not treat communication behavior accurately, as described in this paper. One solution is to model I/O operations as open operations. This abstraction is elegant but generates many spurious behaviors [4,7].

A more general solution to this problem is to lift the power of a model checker from process level to OS level. This way, the effect of an I/O operation is still visible inside the model checker. An existing system that indeed stores and restores full OS states is based on user-mode Linux [10]. That model checker uses the GNU debugger to store states and intercept system calls. The effects of system calls are modeled by hand, but applications can be model checked together without modifying the application code. In that approach, the combined state space of all processes is explored. Our approach analyzes a single process at a time inside a model checker, while running other processes normally. Our approach is therefore more scalable but requires programs to fulfill certain restrictions.

Other virtual machines with such extended replay capabilities have existed before. Initial implementations replayed executions through checkpointing and logging but could not handle system calls [14]. More recent implementations can replay system calls without executing them twice by storing the entire state, including processor registers, before and after such a call [9,12]. They are intended for manual use in conjunction with a debugger, in order to replay one sequence of events. Therefore, they do not allow for systematic state space exploration of a program, which explores a set of states. Furthermore, they use a different approach, intercepting communication at device level, where the network device itself is wrapped. We intercept communication at system call level.

Conventional software model checkers analyze a single process. The results of communication lie outside the scope of such model checkers. Therefore, they cannot be used to model check multi-process applications. One way to solve this problem is to *centralize* a distributed application, i.e., to transform processes into threads [11]. This allows several processes to run in the same model checker, but does not solve the problem of modeling inter-process communication (input/output). Recent work modeled network communication in the centralized model where all processes are executed inside the model checker [1,3].

Centralization model checks multiple communicating processes. However, it may not be possible to execute all processes inside one model checker. To our knowledge, our work is the first approach that allows I/O operations to be carried out by a model checker while still limiting the scope of the model checker to a single process. Furthermore, it even allows model checking of applications where external processes are not running on a platform that the model checkers supports. For example, a server may be written in Java, but clients may be written in a different programming language. While clients cannot run in the model checker VM of JNuke, the server can still be model checked.

6 Conclusions and Future Work

With traditional approaches for model checking software, input/output operations had to be removed by abstraction. Directly resetting program states and executing different branches of a non-deterministic decision is not applicable for communication, as it interacts with the environment of the program and external applications.

In order to solve this problem, a special rollback semantics for stream-based I/O was defined, which includes network communication. If program behavior is independent of the execution schedule, such a program can be model checked using our caching layer semantics. For protocols that do not require repeated interaction, this semantics was successfully implemented in the JNuke model checking engine.

Future work includes possible relaxations of the completeness criteria defined, regarding the order of I/O operations. Specifically, certain interleaved actions should be allowed, such as log entries.

References

1. Artho, C., Garoche, P.: Accurate centralization for applying model checking on networked applications. In: Proc. 21st Int'l Conf. on Automated Software Engineering (ASE 2006), Tokyo, Japan (2006)
2. Artho, C., Schuppan, V., Biere, A., Eugster, P., Baur, M., Zweimüller, B.: JNuke: Efficient Dynamic Analysis for Java. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, Springer, Heidelberg (2004)
3. Artho, C., Sommer, C., Honiden, S.: Model checking networked programs in the presence of transmission failures. In: Proc. 1st Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE 2007), Shanghai, China (2007)
4. Ball, T., Podelski, A., Rajamani, S.: Boolean and Cartesian Abstractions for Model Checking C Programs. In: Margaria, T., Yi, W. (eds.) ETAPS 2001 and TACAS 2001. LNCS, vol. 2031, Springer, Heidelberg (2001)
5. Clarke, E., Grumberg, O., Peled, D.: Model checking. MIT Press, Cambridge (1999)
6. Corbett, J., Dwyer, M., Hatcliff, J., Pasareanu, C., Robby., Laubach, S., Zheng, H.: Bandera: Extracting finite-state models from Java source code. In: Proc. 22nd Intl. Conf. on Software Engineering (ICSE '00), Ireland, ACM Press, New York (2000)
7. Godefroid, P.: Model checking for programming languages using VeriSoft. In: Proc. 24th ACM Symposium on Principles of Programming Languages (POPL '97), France, pp. 174–186. ACM Press, New York (1997)
8. Goldberg, I., Wagner, D., Thomas, R., Brewer, E.: A secure environment for untrusted helper applications. In: Proc. 6th Usenix Security Symposium, San Jose, USA (1996)
9. King, S., Dunlap, G., Chen, P.: Debugging operating systems with time-traveling virtual machines. In: Proc. USENIX 2005 Annual Technical Conference, Anaheim, USA, pp. 1–15 (2005)
10. Nakagawa, Y., Potter, R., Yamamoto, M., Hagiya, M., Kato, K.: Model checking of multi-process applications using SBUML and GDB. In: Proc. Workshop on Dependable Software: Tools and Methods, Yokohama, Japan, pp. 215–220 (2005)
11. Stoller, S., Liu, Y.: Transformations for model checking distributed Java programs. In: Dwyer, M.B. (ed.) Model Checking Software. LNCS, vol. 2057, pp. 192–199. Springer, Heidelberg (2001)
12. Virtutech. Simics Hindsight (2005), `http://www.virtutech.se/products/simics-hindsight.html`
13. Visser, W., Havelund, K., Brat, G., Park, S., Lerda, F.: Model checking programs. Automated Software Engineering Journal 10(2) (April 2003)
14. Zelkowitz, M.: Reversible execution. Commun. ACM 16(9), 566 (1973)

Ant Colony Optimization for Model Checking

Enrique Alba and Francisco Chicano

GISUM Group, Departamento de Lenguajes y Ciencias de la Computación
University of Málaga, Spain
eat@lcc.uma.es, chicano@lcc.uma.es

Abstract. Model Checking is a well-known and fully automatic technique for checking software properties, usually given as temporal logic formulae on the program variables. Most model checkers found in the literature use exact deterministic algorithms to check the properties. These algorithms usually require huge amounts of computational resources if the checked model is large. We propose here the use of Ant Colony Optimization (ACO) to refute safety properties in concurrent systems. ACO algorithms are stochastic techniques belonging to the class of metaheuristic algorithms and inspired by the foraging behaviour of real ants. The results state that ACO algorithms find optimal or near optimal error trails in faulty concurrent systems with a reduced amount of resources, outperforming algorithms that are the state-of-the-art in model checking. This fact makes them suitable for checking safety properties in large concurrent systems, in which traditional techniques fail to find errors because of the model size.

1 Introduction

From the very beginning of computer research, computer engineers have been interested in techniques allowing them to know if a software module fulfils a set of requirements (specification). These techniques are especially important in critical software, such as airplane or spacecraft controllers, in which people's lives depend on the software system. Modern non-critical software is very complex and these techniques have become a necessity in most software companies.

Model checking [1] is a well-known and fully automatic formal method in which all the possible program states are analyzed (in an explicit or implicit way) in order to prove (or refute) that the program satisfies a given property. This property is specified using a temporal logic like Linear Temporal Logic (LTL) or Computation Tree Logic (CTL). One of the best known explicit model checkers is SPIN [2], which takes a software model codified in Promela and a property specified in LTL as inputs. SPIN transforms the model and the negation of the LTL formula into Büchi automata in order to perform the synchronous product of them. The resulting product automaton is explored to search for a cycle of states containing an accepting state reachable from the initial state. If such a cycle is found, then there exists at least one execution of the system not fulfilling the LTL property (see [3] for details). If such kind of cycle does not exist then the system fulfils the property and the verification ends with success.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 523–530, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The amount of states of the product automaton is very high even in the case of small systems, and it increases exponentially with the size of the model. This fact is known as *the state explosion problem* and limits the size of the model that a model checker can verify. This limit is reached when it is not able to explore more states due to the absence of free memory. Although large models cannot be verified due to the state explosion, we can find errors in them. That is, we cannot ensure that the model is correct (if it is) but we can find an error if it is not correct. We use the expression *refute a property* in this article to refer to the action of finding a model execution violating that property.

Here we propose the use of Ant Colony Optimization (ACO) [4] for finding counterexamples (refuting) of LTL formulae in concurrent systems. ACO algorithms belong to the metaheuristic class of algorithms [5], which are able to find near optimal solutions using a reasonable amount of resources. For this reason, they can be suitable for searching accepting states in the graph of large system models, for which traditional exploration algorithms fail.

The paper is organized as follows. In the next section we present previous algorithms used for explicit state model checking. We focus from the beginning on SPIN because the model checker we use is based on it. In Section 3 the ACO metaheuristic used in this paper is described. In Section 4 we present some experimental results comparing the ACO-based algorithms against traditional exact algorithms for explicit state model checking. Finally, Section 5 outlines the conclusions and future work.

2 Background

The properties that can be specified with LTL formulae can be classified into two groups: *safety* and *liveness* properties [6]. Safety properties can be expressed as assertions that must be fulfilled by all the states of the model, while liveness properties refer to assertions that must be fulfilled by execution paths in the model. For the verification of a general LTL formula in explicit state model checking it is necessary to search for a cycle in the state graph with at least one accepting state. Furthermore, such a cycle must be reachable from the initial state. If this cycle is found the model violates the LTL property. For this task, SPIN uses the Nested Depth First Search algorithm (Nested-DFS) [7].

However, in the particular case of safety properties it is enough to find a single accepting state in the product Büchi automaton. This means that safety properties verification can be transformed into a search for one objective node (one accepting state) in a graph (Büchi automaton). Furthermore, the path from one initial node to one objective node represents an execution of the concurrent system in which the given safety property is violated: an error trail. Short error trails in concurrent systems are preferred to long ones. The reason is that a human programmer analyzing the error trail can understand a short trail in less time than a long one.

The simplification of the graph exploration when dealing with safety properties has been used in previous works to verify safety properties using classical

algorithms in the graph exploration domain. Edelkamp, Lluch-Lafuente, and Leue [8] applied Depth First Search (DFS) and Breadth First Search (BFS) to the problem of verifying safety properties using SPIN. They also used heuristic search for this task in their own tool called HSF-SPIN, an extension of SPIN. In order to perform a heuristic search they assign to every state a heuristic value that depends on the property to verify. They applied classical algorithms for graph exploration such as A*, Weighted A*, Iterative Deepening A*, and Best First Search. The results show that, by using heuristic search, the length of the counterexamples can be shortened and the amount of memory required to obtain an error trail is reduced, allowing the exploration of larger models.

Genetic Algorithms (GAs) have also been applied to the problem of refuting safety properties in concurrent systems. In an early proposal, Alba and Troya [9] used GAs for detecting deadlocks, useless states, and useless transitions in communication protocols. To the best of our knowledge, this is the first application of a metaheuristic algorithm to model checking. Later, Godefroid and Kurshid [10], in an independent work, applied GAs to the same problem using a similar encoding of the paths in the chromosome.

In the present work we propose the utilization of another metaheuristic algorithm: Ant Colony Optimization. Unlike GA, ACO is a metaheuristic designed for searching short paths in graphs. This makes ACOs very suitable for the problem at hand. In order to guide the search we use the same heuristic functions defined by Edelkamp et al. [8]. In fact, we have extended their tool, HSF-SPIN, in order to include our ACO algorithm. In this way, we can use all the heuristic functions implemented in HSF-SPIN and, at the same time, all the existing work related to parsing Promela models and interpreting them.

3 Ant Colony Optimization

ACO [4] is a metaheuristic algorithm inspired by the foraging behaviour of real ants to solve combinatorial optimization problems. The main idea consists in simulating the ant's behaviour in a graph in order to search for the shortest path from an initial node to an objective one. The cooperation among the different simulated ants is a key factor in the search. This cooperation is performed indirectly by means of the *pheromone trails*, which is a model of the chemicals the real ants use for their communication. In the following we will describe the different components of an ACO algorithm. But, before that, we have to introduce the combinatorial optimization problems.

A combinatorial optimization problem can be represented by a triplet (S, f, Ω), where S is the set of candidate solutions, f is the *fitness function* that assigns a real value to each candidate solution related to its quality, and Ω is a set of constraints that the final solution must fulfil. The objective is to find a solution minimizing or maximizing the function f (in the following we assume that we deal with minimization problems). A candidate solution is represented by a sequence of *components* chosen from a set C.

In ACO, there is a set of artificial ants (colony) that build the solutions using a stochastic constructive procedure. In the construction phase, ants walk randomly on a graph $G = (C, L)$ called *construction graph*, where L is the set of *connections* (arcs) among the components (nodes) of C. In general, the construction graph is fully connected (is complete), however, some of the problem constraints (elements of Ω) can be modelled by removing arcs from L. Each connection l_{ij} has an associated pheromone trail τ_{ij} and can also have an associated heuristic value η_{ij}. Both values are used to guide the stochastic construction phase that ants perform. However, pheromone trails are modified by the algorithm along the search whilst heuristic values are established from external sources (the designer). Pheromone trails can also be associated to graph nodes (solution components) instead of arcs (component connections). This variation is specially suitable for problems in which the order of the components is not relevant (e.g., subset problems [11]).

```
procedure ACOMetaheuristic
    ScheduleActivities
        ConstructAntsSolutions
        UpdatePheromones
        DaemonActions // optional
    end ScheduleActivities
end procedure
```

Fig. 1. Pseudo-code of the ACO Metaheuristic

In Fig. 1 we reproduce a general ACO pseudo-code found in [4]. It consists of three procedures executed during the search: `ConstructAntsSolutions`, `UpdatePheromones`, and `DaemonActions`. They are executed until a given stopping criterion is fulfilled, such as finding a solution or reaching a given number of steps. In the first procedure each artificial ant follows a path in the construction graph. The ant starts in an initial node and then it stochastically selects the next node according to the pheromone and the heuristic value associated with each arc (or the node itself). The ant appends the new node (component) to the traversed path and selects the next node in the same way. This process is iterated until a candidate solution is built. In the `UpdatePheromones` procedure, pheromone trails associated to arcs are modified. Finally, the last (optional) procedure `DaemonActions` performs centralized actions that are not performed by individual ants. For example, a local optimization algorithm can be implemented in this procedure in order to improve the tentative solution held in every ant.

In this work we use a kind of ACO algorithm called Max-Min Ant System ($\mathcal{MM}$AS) in which the pheromone values are bounded by two values: τ_{min} and τ_{max}. During the construction phase, when ant k is in node i it selects node j with probability

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \text{ if } j \in N_i \ , \tag{1}$$

where N_i is the set of successor nodes for node i, and α and β are two parameters of the algorithm determining the relative influence of the heuristic value and the pheromone trail on the path construction, respectively. Since we are dealing with very large graphs we cannot let the ants walk until they find an accepting state because they can use all the memory for building just one solution. We establish a maximum length for the paths that the ants build, that is, a maximum depth in the exploration of the Büchi automaton. This length is called λ_{ant}.

After the construction phase the pheromone trails are multiplied by $(1 - \rho)$, where ρ is the *pheromone evaporation rate*, which holds $0 < \rho \leq 1$. Then the pheromone trails are updated in order to take into account the quality of the candidate solutions built by the ants. This update follows the expression

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^{bs}, \ \forall (i,j) \in L \ , \tag{2}$$

where $\Delta\tau_{ij}^{bs}$ is the amount of pheromone that the best ant path ever found deposits on arc (i, j). This quantity is usually in direct relation with the quality of the solution. In our case, we try to minimize an objective function (the fitness function) and thus we set $\Delta\tau_{ij}^{bs}$ to the inverse of the minimum fitness value found.

Finally, as we mentioned above, in $\mathcal{MM}$AS the value of the pheromone trails are bounded by τ_{min} and τ_{max}. The values of the trail limits are

$$\tau_{max} = \frac{Q}{\rho} \ , \tag{3}$$

$$\tau_{min} = \frac{\tau_{max}}{a} \ , \tag{4}$$

where Q is the inverse of the minimum fitness value found. The parameter a controls the size of the interval. When one pheromone trail is greater than τ_{max} it is set to τ_{max} and, in a similar way, when one pheromone trail is lower than τ_{min} it is set to τ_{min}. Each time a new better solution is found the trail limits are updated consequently and all pheromone trails are checked in order to keep them inside the interval.

4 Experimental Section

In this section we present some results obtained with the ACO algorithms. For the experiments we have selected five Promela models previously reported in the literature by Edelkamp et al. [12]. All these models have a deadlock state, that is, for each model there exists an execution that leads to a deadlock state. In Table 1 we present the models with some information about them. They can be found in `http://www.albertolluch.com` with the HSF-SPIN source code.

Out from these models, the largest ones are `garp` and `marriers4`. They have a large associated Büchi automata which do not fit in the main memory of any regular machine. The first model, `basic_call2`, simulates a phone call between two peers. The second model, `garp`, implements the Generic Attribute Registration Protocol. The `giop12` model is an implementation of the CORBA Inter-ORB

Table 1. Promela models used in the experiments

Model	LoC	States	Processes
basic_call2	198	33667	3
garp	272	unknown	8
giop12	717	27877	10
marriers4	142	unknown	5
phi8	34	6561	9

protocol for 1 client and 2 servers. `marriers4` is a protocol solving the stable marriage problem for 4 suitors. The Dijkstra dining philosophers problem is implemented in `phi8` with 8 philosophers.

The $\mathcal{MM}$AS algorithms used in the experiments have a colony size of 10 ants and perform 10 steps of the algorithm main loop. The parameters a, ρ, and α are set to 5, 0.1, and 1.0, respectively. The length of the ant paths (λ_{ant}) is set to the double of the optimum (shortest) length to an accepting state. We use two versions of $\mathcal{MM}$AS. In the first one, called $\mathcal{MM}$AS-b, there is no heuristic information for guiding the ants in the construction phase (so the parameter β has no sense). We compare this version against Depth-First Search (DFS) and Bread-First Search (BFS) algorithms, which perform also a blind search on the exploration graph and are the most used algorithms in current model checkers. In our second $\mathcal{MM}$AS algorithm, called $\mathcal{MM}$AS-h, the number of enabled transitions in each state is used as heuristic value for guiding the ants (see [8]) and we set $\beta = 1.0$ in this case. We compare this version against the A* algorithm using the same heuristic. The fitness value (to minimize) of a solution is the sum of the solution length and the heuristic value of the last state. The stopping criterion used in our $\mathcal{MM}$AS algorithms is to find an error trail or to reach the maximum number of allowed steps (10).

Since $\mathcal{MM}$AS is a stochastic algorithm, we need to perform several independent runs in order to get an idea of the behaviour of the algorithm. For this reason we perform 100 independent runs in order to get a high statistical confidence. In all the independent runs $\mathcal{MM}$AS-b and $\mathcal{MM}$AS-h found an error trail. The machine used in the experiments is a Pentium IV at 2.8 GHz with 512 MB of RAM. In Table 2 we show the average length of the error trails found, the average CPU time used, and the average total amount of memory used for all the algorithms.

We can observe in the results that our $\mathcal{MM}$AS algorithms are the unique ones in finding out a deadlock state in `marriers4` model. The exact algorithms (DFS, BFS, and A*) were stopped by the operative system because of their memory consumption and thus were unable of discovering the deadlock state. Among the algorithms without heuristic information, $\mathcal{MM}$AS-b obtains error trails of a similar length as BFS but using much less resources (memory and CPU time). The length of the error trails found by DFS are much longer (bad quality) than those of the other two algorithms (BFS and $\mathcal{MM}$AS-b).

Comparing the algorithms using heuristic information we can observe that $\mathcal{MM}$AS-h gets near optimal error trails with important savings of resources

Table 2. Results obtained with all the algorithms

Concurrent System		DFS	BFS	$\mathcal{MM}$AS-b	A*	$\mathcal{MM}$AS-h
`basic_call2`	Length	82.00	**26.00**	30.24	**26.00**	31.30
Basic call protocol	CPU (ms)	**10.00**	80.00	41.70	110.00	**49.40**
with 2 users	Mem. (KB)	**2713.00**	16384.00	4219.44	17408.00	**4278.44**
`garp`	Lenth	65.00	**17.00**	24.70	**17.00**	24.00
Generic Attribute	CPU (ms)	10.00	53180.00	**6.00**	2820.00	**9.50**
Registration Protocol	Mem. (KB)	3357.00	480256.00	**2532.28**	122880.00	**2634.44**
`giop12`	Length	48.00	**43.00**	43.45	**43.00**	**43.00**
CORBA General Inter-ORB	CPU (ms)	**10.00**	350.00	41.20	490.00	**26.80**
Protocol (1 client, 2 servers)	Mem. (KB)	**2901.00**	49152.00	4085.12	37888.00	**3202.24**
`marriers4`	Length	-	-	**85.79**	-	**88.76**
Stable marriage problem	CPU (ms)	-	-	**148.60**	-	**91.90**
with 4 suitors	Mem. (KB)	-	-	**15388.39**	-	**10625.80**
`phi8`	Length	1338.00	**10.00**	**10.00**	**10.00**	**10.00**
Dining philosophers	CPU (ms)	70.00	60.00	**21.90**	**0.00**	35.20
with 8 philosophers	Mem. (KB)	29696.00	17408.00	**4333.92**	**2105.00**	4986.68

with respect to A*. Furthermore, in the `giop12` model, $\mathcal{MM}$AS-h obtains in the 100 independent runs an error trail with the minimum length using one tenth of the memory and one twentieth of the time that A* requires for the same goal.

The results above state that ACO algorithms find optimal or near optimal error trails in faulty concurrent systems using a reduced amount of resources, outperforming algorithms that are the state of the art in model checking. This fact makes them suitable for checking safety properties in large concurrent systems, in which traditional techniques fail to find errors because of the model size.

5 Conclusions and Future Work

We have presented here a novel application of Ant Colony Optimization to the problem of finding counterexamples of safety properties in concurrent systems. We have compared the ACO algorithms against the state-of-the-art exhaustive methods and the results show that ACO algorithms are able to outperform the state-of-the-art algorithms in efficacy and efficiency. They require a very low amount of memory and CPU time and are able to find errors even in models in which the state-of-the-art algorithms fail because of the high amount of memory required during the search.

ACO algorithms can be used with other techniques for reducing the amount of memory required in the search such as partial order reduction, symmetry reduction, or state compression. As a future work we plan to combine these techniques with ACO. We also plan to extend the application of ACO to the search for liveness properties violations. This can be done in two phases. First, the algorithm searches an accepting state and then it tries to find a path to this state from itself.

In the present paper it has been shown that the way in which ACO performs the search is very useful for finding design errors. However, ACO algorithms do not ensure the correctness of the models in which no error is found. We plan to use ACO algorithms as the base of an exhaustive stochastic algorithm that is

able to ensure the correctness when no error is found while at the same time is able to find errors very fast when they exist.

Model checkers working in parallel on a cluster of machines are gaining importance in the formal methods community nowadays. In addition, a lot of work exists stating the high efficiency and efficacy of parallel metaheuristics [13]. We plan to design a parallel version of ACO for reducing the time required and increasing the available memory, thus able to work with even larger models. We also want to integrate the algorithm inside Java PathFinder, which is able to work with programs in Java language, much more familiar for the computer science community than Promela.

Acknowledgements. This work has been partially funded by the Ministry of Education and Science and FEDER under contract TIN2005-08818-C04-01 (the OPLINK project). Francisco Chicano is supported by a grant (BOJA 68/2003) from the Junta de Andalucía.

References

1. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. The MIT Press, Cambridge (2000)
2. Holzmann, G.J.: The model checker SPIN. IEEE Transactions on Software Engineering 23(5), 1–17 (1997)
3. Holzmann, G.J.: The SPIN Model Checker. Addison-Wesley, Reading (2004)
4. Dorigo, M., Stützle, T.: Ant Colony Optimization. The MIT Press, Cambridge (2004)
5. Blum, C., Roli, A.: Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Computing Surveys 35(3), 268–308 (2003)
6. Manna, Z., Pnueli, A.: The temporal logic of reactive and concurrent systems. Springer, New York (1992)
7. Holzmann, G.J., Peled, D., Yannakakis, M.: On nested depth first search. In: Proc. Second SPIN Workshop, American Mathematical Society, pp. 23–32 (1996)
8. Edelkamp, S., Lafuente, A.L., Leue, S.: Directed Explicit Model Checking with HSF-SPIN. In: Dwyer, M.B. (ed.) Model Checking Software. LNCS, vol. 2057, pp. 57–79. Springer, Heidelberg (2001)
9. Alba, E., Troya, J.: Genetic Algorithms for Protocol Validation. In: Proceedings of the PPSN IV International Conference, pp. 870–879. Springer, Berlin (1996)
10. Godefroid, P., Khurshid, S.: Exploring very large state spaces using genetic algorithms. Intl. Jnl. on Software Tools for Technology Transfer 6(2), 117–127 (2004)
11. Leguizamón, G., Michalewicz, Z.: A new version of Ant System for subset problems. In: Angeline, P., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A. (eds.) Proceedings of the 1999 Congress on Evolutionary Computation, pp. 1459–1464. IEEE Computer Society Press, Piscataway, New Jersey (1999)
12. Edelkamp, S., Leue, S., Lluch-Lafuente, A.: Directed explicit-state model checking in the validation of communication protocols. International Journal of Software Tools for Technology Transfer 5, 247–267 (2004)
13. Alba, E. (ed.): Parallel Metaheuristics. A New Class of Algorithms. Wiley, Chichester (2005)

On Combining 01X-Logic and QBF*

Marc Herbstritt and Bernd Becker

Albert-Ludwigs-University, Freiburg im Breisgau, Germany
{herbstri,becker}@informatik.uni-freiburg.de

Abstract. We discuss how to combine 01X-logic and quantified boolean formulas (QBF) within a homogeneous SAT/QBF-framework in the context of bounded model checking of blackbox designs. The proposed combination allows a flexible handling of blackboxes w.r.t. computational resources. Preliminary results show the scalability of the approach.

Keywords: 01X, QBF, Bounded Model Checking, Blackbox Designs.

1 Introduction

A SAT-based approach for bounded model checking (BMC) of blackbox designs requires an adequate modeling of the blackbox behavior. The most simple modelling style relies on 01X-logic where the third logical value X represents the unknown behavior of the blackbox. As in standard BMC [1], for 01X-logic a boolean formula stating the existence of a counterexample (CE) at unfolding depth k can be generated. The CEs found by this approach are *independent* of the blackbox behavior. CEs that *depend* on the blackbox behavior can be described by a formula using quantifiers, resulting in quantified boolean formulas (QBFs), where especially the universal quantifier ($\forall$) allows to express that CEs have to be valid *for every possible* behavior of the blackbox [2].

In this work we investigate, how 01X-logic and QBF can be combined such that a homogeneous SAT/QBF-based decision procedure can be applied. The paper is structured as follows. After discussing briefly related work in Sect. 2, preliminaries are given in Sect. 3. In Sect. 4, we show how to combine 01X-logic and QBF along a small example. The QBF formulation for BMC of blackbox designs is briefly revisited in Sect. 5, and preliminary experimental results for this application are presented in Sect. 6. The paper is concluded in Sect. 7.

2 Related Work

Bounded model checking of blackbox designs was initially proposed in [3] by using 01X-logic, and was further extended in [2] to a more concise QBF formulation. The two-valued encoding approach for 01X-logic on which we rely on was presented in [4]. In the context of BDD-based symbolic model checking of blackbox designs, handling the blackboxes in different styles was analyzed in [5].

* This work was partly supported by the German Research Council (DFG) as part of SFB/TR 14 AVACS "Automatic Verification and Analysis of Complex Systems" (www.avacs.org).

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 531–538, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Table 1. Boolean operators AND, OR, and NOT extended to 01X-logic

$\texttt{AND}_{01X}(a,b)$			
a \ b	0	1	X
0	0	0	0
1	0	1	X
X	0	X	X

$\texttt{OR}_{01X}(a,b)$			
a \ b	0	1	X
0	0	1	X
1	1	1	1
X	X	1	X

$\texttt{NOT}_{01X}(a)$	
a	
0	1
1	0
X	X

3 Preliminaries

We review 01X-logic and how it can be reduced to propositional logic using a binary encoding. We also briefly introduce Quantified Boolean Formulas (QBFs).

01X-Logic. 01X-logic extends propositional logic by a third logical value, X, to provide means to talk about the uncertainty of the status of propositional variables. Basic operators like conjunction, disjunction, and negation can be adapted conservatively by taking the controlling values of the gate function into account, as it is shown in Table 1. In [4], Jain et al. propose an approach for handling the three logical values $0, 1$, and X by a binary encoding to which the operators can be extended.

Definition 1. *The three logical values* $0_{01X}, 1_{01X}$, *and* X_{01X} *are binary encoded as:*[1] $0_{01X} := (1,0)$, $1_{01X} := (0,1)$, *and* $X_{01X} := (0,0)$. *The operators can be adapted to this encoding scheme, now using tuples* (a_0,a_1) *and* (b_0,b_1) *for 01X-variables:* $\texttt{AND}_{01X}((a_0,a_1),(b_0,b_1)) := (a_0+b_0, a_1 \cdot b_1)$, $\texttt{OR}_{01X}((a_0,a_1),(b_0,b_1)) := (a_0 \cdot b_0, a_0+b_0)$, *and* $\texttt{NOT}_{01X}((a_0,a_1)) := (a_1,a_0)$.

Quantified Boolean Formulas. Quantified Boolean Formulas (QBFs) allow existential ($\exists$) and universal ($\forall$) quantification of boolean variables. A QBF of the form $\exists x : f(x)$ ($\forall x : f(x)$) is called *valid* iff for some (all) value(s) of $x \in \{0,1\}$, $f(x)$ evaluates to 1. Briefly, a QBF has the following general form:

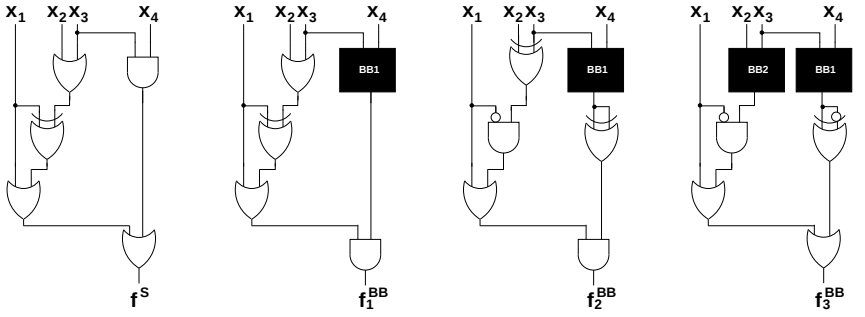
$$Q_1 X_1 Q_2 Y_1 Q_3 X_2 Q_4 Y_2 \ldots Q_n Y_{\frac{n}{2}} Q_{n+1} X_{\lceil \frac{n+1}{2} \rceil} : f(X_1, Y_1, X_2, Y_2, \ldots, Y_{\frac{n}{2}}, X_{\lceil \frac{n+1}{2} \rceil}),$$

where $X_1, Y_1, \ldots, Y_{\frac{n}{2}}, X_{\lceil \frac{n+1}{2} \rceil}$ are pairwise disjoint sets of propositional variables, $Q_1 = \exists, Q_i \in \{\exists, \forall\}, Q_i \neq Q_{i+1}, 1 \leq i \leq n$, and f is a propositional formula in *Conjunctive Normal Form* (CNF).

4 Towards Combining 01X-Logic and QBF: An Example

We apply 01X-logic as well as a QBF formulation to a small example for combinational equivalence checking (CEC) of blackbox designs, i.e., partial circuit implementations. The example is analyzed step by step, first showing that 01X-logic is able to detect errors. Second, we show that 01X-logic is sometimes not

[1] 0 and 1 are used instead of 0_{01X} and 1_{01X}, when the meaning is clear from the context.



(a) f^S **(b)** f_1^{BB} **(c)** f_2^{BB} **(d)** f_3^{BB}

Fig. 1. Specification circuit f^S, blackbox designs f_1^{BB} and f_2^{BB} with one blackbox, and blackbox design f_3^{BB} with two blackboxes

expressive enough and that a QBF formulation has to be used to detect errors. Finally, we present an example to show how a combination of 01X-logic and QBF can be implemented in the presence of multiple blackboxes, and that it is still possible to detect design errors.

Detecting errors using 01X-Logic. Our running example is about CEC, i.e., to prove or disprove equivalence of boolean functions. In Fig. (1a) the specification circuit f^S, that is used throughout this paper, is depicted. We are interested to find errors of blackbox designs, e.g., the one depicted in Fig. (1b). The equivalence checking problem now has to analyze whether the circuit of f_1^{BB} already contains an error compared to f^S. This is done by constructing a so-called *miter* [6] that connects corresponding outputs of the specification and implementation by an `XOR`-gate such that a logical value 1 at the miter output indicates a difference, i.e., $f_M(f^S, f_1^{\mathrm{BB}}) = f^S \oplus f_1^{\mathrm{BB}}$. An input assignment $(x_1, x_2, x_3, x_4)_{\mathrm{CE}}$ to the miter circuit that leads to such a difference, i.e., $f_M(f^S, f_1^{\mathrm{BB}})(x_1, x_2, x_3, x_4)_{\mathrm{CE}} = 1$, is a *counterexample* (CE). The unknown behavior of the blackbox is modelled by removing the blackbox from the design and replacing the blackbox output by the logical value X of 01X-logic. For the analysis whether there is a CE that is observable already by using 01X-logic, we transform the miter circuit for f_M using the binary encoding of Def. 1, resulting in a tuple (f_M^0, f_M^1). Please note that the logical value of X is replaced by the tuple $(0, 0)$. Since we require a logical value 1 (that is mapped to $(0, 1)$) at the miter output, we have to check whether the boolean function $((\neg f_M^0) \cdot f_m^1)$ is satisfiable. This is done by transforming the miter circuit into CNF using additional Tseitin-variables [7,8]. Let T be the set of auxiliary Tseitin-variables and let $(((\neg f_M^0) \cdot f_m^1) = 1)_{\mathrm{CNF}}$ be the CNF encoding of $((\neg f_M^0) \cdot f_m^1) = 1$, then we have to check the satisfiability of the propositional formula

$$\exists\, x_1^0 x_1^1\, x_2^0 x_2^1\, x_3^0 x_3^1\, x_4^0 x_4^1\; T : (((\neg f_M^0) \cdot f_m^1) = 1)_{\mathrm{CNF}}, \tag{1}$$

which can be perfomed by using off-the-shelf SAT solvers, e.g., [9,10,11]. Regarding f_S and f_1^{BB}, this method would reveal that the input assignment $(0,0,0,1)$ is a CE, since $f^S(0,0,0,1) = 0 \neq 1 = f_1^{\mathrm{BB}}(0,0,0,1)$.

Inaccuracy of 01X-logic and the demand for QBF. The application of 01X-logic is a fairly coarse grained abstraction for the unknown behavior of the blackboxes. Let's have a look at the blackbox design f_2^{BB} given in Fig. (1c). By using 01X-logic, it cannot be detected that the output of the `XOR`-gate, that is driven by the blackbox output, is constantly 0. Hence, no CE can be found by 01X-logic. Instead, we can introduce a fresh variable Z and replace the blackbox output by this additional variable. Now, a concrete CE requires that *for all* values of variable Z, a difference at the outputs of f^S and f_2^{BB} can be observed. Hence, this approach leads to a QBF formulation instead of a pure propositional formula, and we have to check the validity of the following formula (taking into account the CNF-transformation for the propositional kernel $f_M(f^S, f_2^{\mathrm{BB}})(x_1, x_2, x_3, x_4, Z) = 1$ and the set T of Tseitin-variables that comes along with it):

$$\exists\, x_1\, x_2\, x_3\, x_4\, \forall Z\, \exists T\ :\ f_M(f^S, f_2^{\mathrm{BB}})_{\mathrm{CNF}}. \tag{2}$$

The validity of formula (2) can be checked by generic QBF solvers, e.g., [12,13,14]. We denote this approach *pure Z_i-QBF* in the following, since for several blackbox outputs, several variables Z_i are introduced. Fortunately, the pure Z_i-QBF is able to detect an error for f_2^{BB}. E.g., from a certificate that states that formula (2) is valid, we can extract that the input assignment $(0,1,1,0)$ is a CE.

Handling of Multiple Blackboxes. The previous blackbox designs only had one blackbox. But depending on the application context, there may be multiple blackboxes, see e.g. Fig. (1d). This example is slightly different to f_2^{BB}, because here the output of the blackbox BB_1 is connected to a gate whose output has constantly the value 1. Clearly, this error can be detected although there are two blackboxes in the partial implementation. Using the pure Z_i-QBF approach, the existence of at least one CE can be proven by checking the validity of the formula

$$\exists\, x_1\, x_2\, x_3\, x_4\, \forall Z_1\, Z_2\, \exists T\ :\ f_M(f^S, f_3^{\mathrm{BB}})_{\mathrm{CNF}}.^2 \tag{3}$$

Is it possible to apply the binary encoding to formula (3)? The answer to this question brings us one step closer to the main topic of our paper, namely how to combine 01X-logic and QBF. For the propositional part of formula (3) there is no problem at all for applying the binary encoding. But what about the quantifiers $\forall Z_1$ and $\forall Z_2$? In a first shot, we could write the following:

$$\exists\, x_1^0 x_1^1\, x_2^0 x_2^1\, x_3^0 x_3^1\, x_4^0 x_4^1\, \forall Z_1^0 Z_1^1\, Z_2^0 Z_2^1\, \exists T\ :\ \big(((\neg f_M^0) \cdot f_M^1) = 1\big)_{\mathrm{CNF}}. \tag{4}$$

Is this adequate? No, because universally quantifying *both* variables of the tuple, e.g., $\forall Z_1^0 Z_1^1$, means to check all four propositional combinations, i.e.,

[2] The input assignments $(0,0,0,0)$ and $(0,0,0,1)$ are CEs for f_3^{BB}.

$(0,0),(0,1),(1,0),(1,1)$. But from the viewpoint of 01X-logic, this means to check the values $X_{\texttt{01X}}, 1_{\texttt{01X}}$, and $0_{\texttt{01X}}$ and for the tuple $(1,1)$ there exists no interpretation at all. What goes wrong here is that we should only check the 01X-values $0_{\texttt{01X}}$ and $1_{\texttt{01X}}$ for Z_1, since Z_1 is a *propositional* variable and not a 01X-variable. Hence, the answer for the correct transformation of the quantifiers is found in the encoding of $0_{\texttt{01X}}$ and $1_{\texttt{01X}}$. Using Def. 1, $0_{01\mathrm{X}}$ ($1_{01\mathrm{X}}$) is encoded by $(1,0)$ $((0,1))$. As one can see, the first values of the tuples are complementary. Additionally, the second value of each tuple is always the opposite of the first one. We exploit this observation for a two-stage transformation scheme:

1. Universally quantified variables $\forall Z_i$ have to be replaced by $\forall Z_i^0 \, \exists Z_i^1$.
2. To ensure that only the logical values $0_{01\mathrm{X}}$ and $1_{01\mathrm{X}}$ are checked, we have to add the following two clauses to the CNF:
   (a) $(Z_i^0 \rightarrow (\neg Z_i^1))$ (b) $((\neg Z_i^0) \rightarrow Z_i^1)$.

Hence, the transformation for the binary encoding of formula (3) becomes

$$\begin{aligned} \exists\, x_1^0 x_1^1\, x_2^0 x_2^1\, x_3^0 x_3^1\, x_4^0 x_4^1\, \forall \mathbf{Z_1^0}\, \exists \mathbf{Z_1^1}\, \forall \mathbf{Z_2^0}\, \exists \mathbf{Z_2^1}\, \exists T \, : \\ ((\neg f_M^0) \cdot f_M^1) = 1)_{\mathrm{CNF}} \cdot \\ (Z_1^0 \rightarrow (\neg Z_1^1)) \cdot ((\neg Z_1^0) \rightarrow Z_1^1) \cdot \\ (Z_2^0 \rightarrow (\neg Z_2^1)) \cdot ((\neg Z_2^0) \rightarrow Z_2^1). \end{aligned} \tag{5}$$

Combining 01X-Logic and QBF. Actually, we have already discussed all necessary prerequisites to combine 01X-logic and QBF. The main problem, namely the handling of the universally quantified variables, was solved in the preceding paragraph. Nevertheless, let's have a look at the binary encoded miter circuit for f_3^{BB} where blackbox BB_1 is modelled using an additional variable Z_1 and BB_2 is modelled using 01X-logic. In contrast to formula (5), only variable Z_1 has to be quantified universally. The 01X-encoded output of BB_2 is replaced by the tuple $(0,0)$ that correspond to $X_{\texttt{01X}}$, and hence no variables have to be quantified w.r.t. BB_2. Finally, the QBF for checking the existence of a CE becomes

$$\begin{aligned} &\exists\, x_1^0 x_1^1\, x_2^0 x_2^1\, x_3^0 x_3^1\, x_4^0 x_4^1\, \forall Z_1^0\, \exists Z_1^1\, \exists T \, : \\ &((\neg f_M^0) \cdot f_M^1) = 1)_{\mathrm{CNF}} \cdot (Z_1^0 \rightarrow (\neg Z_1^1)) \cdot ((\neg Z_1^0) \rightarrow Z_1^1). \end{aligned} \tag{6}$$

By modelling BB_2 using 01X-logic, we have reduced the complexity of the QBF, since we got rid of one universally quantified variable, namely Z_2^0 (additionally we also got rid of Z_2^1). Reducing the number of universally quantified variables can be crucial for deciding the validity of a QBF by a QBF-solver. Hence, with the presented scheme we are now able to scale the complexity of the QBF in terms of universally quantified variables, by deciding to model some blackboxes using 01X-logic and some by additional Z_i-variables.[3]

[3] The soundness of the transformation can be proven formally, but is left out due to page limitations.

5 Application to Bounded Model Checking of Blackbox Designs

In [2,3] implementations of decision procedures for deciding 01X-logic in the context of bounded model checking of blackbox designs (BB-BMC) were presented. In [2] also a QBF for BB-BMC is presented that is as follows:

$$\begin{aligned}\varphi_{BMC}(d) := \; & \exists s^1 \, \exists x^1 \, \exists \Xi_{(1,1)} \, \forall \Gamma_{(1,1)} \, \ldots \exists \Xi_{(\beta,1)} \, \forall \Gamma_{(\beta,1)} \\ & \exists s^2 \, \exists x^2 \, \exists \Xi_{(1,2)} \, \forall \Gamma_{(1,2)} \, \ldots \exists \Xi_{(\beta,2)} \, \forall \Gamma_{(\beta,2)} \\ & \ldots \\ & \exists s^{d-1} \, \exists x^{d-1} \, \exists \Xi_{(1,d-1)} \, \forall \Gamma_{(1,d-1)} \, \ldots \exists \Xi_{(\beta,d-1)} \, \forall \Gamma_{(\beta,d-1)} \\ & \exists s^d : \qquad \mathrm{IOC}(\beta, d) \rightarrow \big(\mathrm{I}(s^1) \cdot T(\beta, d) \cdot (\neg P(s^d))\big)\end{aligned} \tag{7}$$

Here, β denotes the number of blackboxes and d is the unfolding depth. Variables $\Xi_{(b,u)}$ ($\Gamma_{(b,u)}$) represent vectors of input (output) variables of blackbox b in time frame u. In formula (7), $I(s^1)$ is a predicate for the initial states, $T(\beta, d)$ is the transition relation describing d transitions, and $P(s^d)$ describes the safety property in time frame d, i.e., $\neg P(s^d)$ means that the state reached after d time steps violates the property P. To avoid false negative results, we require that the combinational behavior of the blackboxes is respected, i.e., when the blackbox inputs are assigned the same values within different time frames, then the values of the blackbox outputs have to be the same. This property is captured by a predicate $\mathrm{IOC}(\beta, d)$[4] that is true iff the consistency constraints for all β-many blackboxes hold in the finite unfolding up to depth d. Now, when formula $\varphi_{BMC}(d)$ is *true*, then there exists a CE.[5] In [2] we found experimentally that the 01X-logic approach performs very well, but on the contrary the QBFs are very hard to solve for current state-of-the-art QBF solvers.[6] One possible way to overcome the empirical complexity of the QBFs related to blackbox designs is to individually decrease the accuracy of the modelling style. The idea is to allow both, 01X-based and Z_i-based blackbox modelling for different blackboxes.

6 Experimental Results

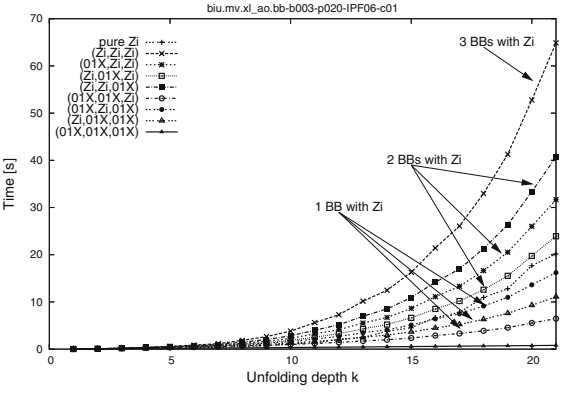
We have applied our transformation scheme to two blackbox designs that contain exactly 3 blackboxes: `biu-mv-xl-ao-bb-b003-p020-IPF06-c01` and `biu-mv-xl- ao-bb-b003-p020-IPF01-c05`.[7] The accuracy of modelling the

[4] IOC = *Input-Output-Consistency*.

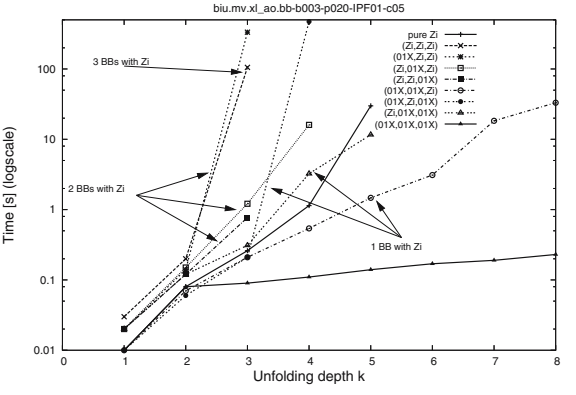
[5] The CE is given in terms of a Q-model (see [14]), such that for every blackbox implementation there exists an individual CE that corresponds to a path of lenght d in the Q-model of $\varphi_{BMC}(d)$.

[6] Only 1 out of 28 QBF-solvers was able to solve all submitted examples in the competitive QBF evaluation in 2006 [15].

[7] The underlying sequential circuits were obtained from the examples coming with the VIS verification suite [16]. The blackboxes and the defects are introduced randomly, see [3,2] for details.



(a) `biu.mv_xl.ao_bb.b003-p020-IPF06-c01`



(b) `biu.mv_xl.ao_bb.b003-p020-IPF06-c01`

Fig. 2. Results for different modelling styles of the 3 blackboxes

blackboxes was increased step-wise, i.e., in the beginning, every blackbox is modelled by 01X, then 1 out of the 3 blackboxes is modelled by Z_i, afterwards 2 out of 3 blackboxes are modelled by Z_i, and finally every blackbox is modelled by Z_i. Additionally, we generated the QBFs for the pure Z_i-QBF approach, i.e., this corresponds to the QBFs generated in [2]. The results for `biu-mv-xl-ao-bb-b003-p020-IPF06-c01`, depicted in Fig. (2a), nicely show how the complexity in terms of CPU time increases as the number of Z_i-modelled blackboxes increases. The pure Z_i-QBFs according to [2] are more easily to solve than those examples where at least two blackboxes are modelled by Z_i and the binary encoding was applied. But in general, the results do not scale as smoothly as for `biu-mv-xl-ao-bb b003-p020-IPF06-c01`. Fig. (2b) shows the results for `biu-mv-xl-ao-bb-b003- p020-IPF01-c05`. Again, the designs where only one blackbox is modelled by Z_i-variables outperform or are competitive to the pure Z_i-QBF approach. When two blackboxes are modelled using Z_i-variables, the

complexity for the binary encoding impacts the efficiency of the underlying QBF-solver, such that the pure Z_i-QBF approach is not outperformed.

7 Conclusions

In this paper we presented an approach for combining 01X-logic and QBF in the context of bounded model checking of blackbox designs. The approach relies on applying a two-valued transformation scheme that takes the quantifier of the variables into account such that correction clauses have to be added to ensure consistency. Experimental results have shown that this approach enables scalability by flexible handling of the blackboxes.

Acknowledgments. The authors would like to thank Tobias Nopper.

References

1. Biere, A., Cimatti, A., Clarke, E., Strichman, O., Zhu, Y.: Bounded Model Checking. Advances in Computers 58 (2003)
2. Herbstritt, M., Becker, B., Scholl, C.: Advanced SAT-Techniques for Bounded Model Checking of Blackbox Designs. In: Proc. of MTV, pp. 37–44. IEEE CS, Los Alamitos (2006)
3. Herbstritt, M., Becker, B.: On SAT-based Bounded Invariant Checking of Blackbox Designs. In: Proc. of MTV, pp. 23–28. IEEE CS, Los Alamitos (2005)
4. Jain, A., et al.: Testing, Verification, and Diagnosis in the Presence of Unknowns. In: VLSI Test Symp., pp. 263–269 (2000)
5. Nopper, T., Scholl, C.: Flexible modeling of unknowns in model checking for incomplete designs. In: GI/ITG/GMM Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (2005)
6. Brand, D.: Verification of large synthesized designs. In: Proc. of ICCAD (1993)
7. Tseitin, G.: On the complexity of derivations in propositional calculus. Studies in Constructive Mathematics and Mathematical Logics (1968)
8. Marques-Silva, J., Sakallah, K.: Boolean Satisfiability in Electronic Design Automation. In: Proc. of DAC, pp. 675–680 (2000)
9. Marques-Silva, J., Sakallah, K.: GRASP: A Search Algorithm for Propositional Satisfiability. IEEE Trans. on Comp. 48(5), 506–521 (1999)
10. Moskewicz, M., Madigan, C., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engeneering an Efficient SAT Solver. In: Proc. of DAC (2001)
11. Lewis, M., Schubert, T., Becker, B.: Multithreaded SAT Solving. In: Proc. of ASP-DAC (2007)
12. Giunchiglia, E., Narizzano, M., Tacchella, A.: QuBE++: An Efficient QBF Solver. In: Hu, A.J., Martin, A.K. (eds.) FMCAD 2004. LNCS, vol. 3312, pp. 201–213. Springer, Heidelberg (2004)
13. Biere, A.: Resolve and Expand. In: Hoos, H.H., Mitchell, D.G. (eds.) SAT 2004. LNCS, vol. 3542, pp. 59–70. Springer, Heidelberg (2005)
14. Samulowitz, H., Bacchus, F.: Using SAT in QBF. In: van Beek, P. (ed.) CP 2005. LNCS, vol. 3709, pp. 578–592. Springer, Heidelberg (2005)
15. Narizzano, M., Pulina, L., Tacchella, A.: QBF Evaluation 2006 (2006) [2006-08-02], Available on-line at `www.qbflib.org/qbfeval`
16. The VIS Group: VIS: A system for verification and synthesis. In: Alur, R., Henzinger, T.A. (eds.) CAV 1996. LNCS, vol. 1102, Springer, Heidelberg (1996)

Model Checking a Video–on–Demand Server Using McErlang

Lars-Åke Fredlund[1,⋆] and Juan José Sánchez Penas[2,⋆⋆]

[1] LSIIS, Facultad de Informática, Universidad Politécnica de Madrid
fred@babel.ls.fi.upm.es
[2] MADS Group, Departamento de Computación, Universidade da Coruña
jjsanchez@udc.es

Abstract. The article describes a method to obtain performance measurements from complex distributed systems using a model checking approach. We illustrate the approach by applying it to a Video–on–Demand application developed in Erlang. To obtain performance measurements concerning e.g. streaming capacity, and identify system bottlenecks, we used the McErlang model checker which implements a large part of the Erlang API. Answers to capacity queries are computed as measures over paths in the system state graph, and the combination of an on-the-fly model checker (not requiring the generation of the complete state graph) with a powerful language (Erlang itself) for expressing correctness claims, made it possible to analyse substantially sized systems.

1 Introduction

Erlang is a programming language developed at Ericsson for implementing telecommunication systems [1]. It provides a functional sub-language, enriched with constructs for process creation and communication via message passing. Moreover Erlang has support for writing distributed programs; *processes* can be distributed over physically separated processing *nodes*. A typical industrial class application mostly developed in Erlang is the AXD 301 ATM switch [3]. The software of such products is typically organised into many modules, which at runtime execute as a dynamically varying number of processes communicating through asynchronous message passing. The highly concurrent and dynamic nature of such software makes it particularly hard to debug and test.

In this article we use a new model checker McErlang for Erlang to model check a complex distributed system, VoDKA [7], a Video–on–Demand server developed at the University of A Coruña. In the next section we describe the essential properties of the McErlang model checker, and section 3 introduces the VoDKA Video–on–Demand server. Section 4 describes a method for extracting performance properties from such as a class of distributed systems, interestingly

⋆ Supported by a RyC grant from the Spanish Ministerio de Educación y Ciencia.
⋆⋆ Supported by projects MEyC TIN2005-08986 and XUGA PGIDIT06PXIC105164 PN.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 539–546, 2007.
© Springer-Verlag Berlin Heidelberg 2007

enough using completely standard model checkers (offering neither an implementation of a real–time semantics, nor probabilistic reasoning). Experimental results from applying the method on VoDKA are documented in section 5, and finally section 6 discusses related work and future plans.

2 McErlang: Functional Model Checking for Erlang

McErlang [5] is a model checker for Erlang programs. The input to the model checker is a program together with a special call-back module also written in Erlang which specifies the behavioral property to be checked (called the *monitor*). The output can be either a positive answer saying that the property holds, or a negative one together with a counterexample. The tool can also be used to generate the state graph of a program; this permits the use of other model checking tools like CADP [6] for the actual property checking.

The main idea behind McErlang is to re-use as much of a normal Erlang programming language implementation as possible, but adding a model checking capability. To achieve this, the tool replaces the part of the Erlang runtime system which implements concurrency and message passing, while still using the runtime system for the evaluation of the sequential part of the input programs.

The model checker has a complex internal state in which the current state of the runtime system is represented. The structure that is maintained by the model checker records the state of all alive processes (their process identifiers, mailboxes, computation state, etc). Moreover the global state kept by the model checker runtime system includes a structure to record process links, information about registered process identifiers, etc.

As Erlang lacks a good reflection capability, a program has to undergo a source-to-source translation before model checking; McErlang does this automatically. Essentially calls to functions with side effects in the module `erlang` (providing the standard API for process operations in Erlang) are substituted with calls to functions in the model checker provided module `evOS` instead (which are side-effect free, except that they modify the internal model checker data structures). In addition process sends and receives are transformed as well.

The model checker implements full linear-temporal logic (LTL) checking. Correctness properties are represented as Büechi automatons (*monitors* coded in Erlang) which are checked using a standard on–the–fly dept–first model checking algorithm [9]. For efficiency, there is a dedicated safety property only checker available. A *monitor* checks whether the correctness property holds of the combination of the new program state and the monitor state. If successful, the monitor returns an updated monitor state (for safety checking). Correctness properties can be implemented, therefore, as finite state machines where depending on the monitor state, actions leading to new states are accepted or not.

During model checking calls are also made to the *abstractions* module for abstracting state spaces or transitions, and with the result a call is made to the *state table* in order to determine if a given (abstracted) state has been seen previously or not. The user of the tool can define *ad hoc* abstractions where the

irrelevant information is not taken into account at the time of distinguishing states. Adapted state tables can also be defined, including the possibility of stopping the generation of new states in a given path depending on the state of the model checker. Two states are the same if the abstractions point to the same element of the state table, taking into account both the state of the monitor and the state of the model checker.

McErlang has built-in support for some Erlang/OTP component behaviours that are used in almost all serious Erlang programs such as the supervisor component (for fault-tolerant applications). and the generic server component (implementing a client–server component). The presence of such high-level components in the model checker significantly reduces the gap between original program and the verifiable model, compared to other model checkers.

3 VoDKA: A Distributed Video–on–Demand Server

A Video–on–Demand (VoD) server is a system where the user can ask for any piece of multimedia content (*media object*) at any time. If the media object is available, the system tries to allocate the resources needed to satisfy that request and stream the object back to the user. VoDKA [7] is a VoD server, originally developed at the University of A Coruña by the MADS research group, and nowadays used by industry. Its main requirements were fault tolerance, massive concurrency, huge storage capacity, scalability, variability and low cost.

Given those requirements, the VoDKA system was conceived as a flexible architecture of specialized components, in charge of tasks like massive storage, caching, filtering or streaming. Components cooperate in an architecture based on the concept of delegation, where a cache-based hierarchical configuration keeps popular content close to the fastest layers.

A linear configuration including multi-level caches, showing the internal processes of each component, can be seen in Fig. 1. The upper part of the figure represents the supervision tree of processes in charge of starting the system and providing fault tolerance. The streaming component offers to the user applications different kinds of front-ends. The cache component is in charge of the devices where the media objects are stored temporarily to increase system performance. The storage component takes care of devices where media objects are stored permanently. One of this storage devices is, in fact, a *wrapper* for another VoDKA system accessed using HTTP. At the bottom of the figure, transfers takes place using temporary processes.

The communication inside and among the components is based on a message API kept as simple and homogeneous as possible. Almost the same messages are sent through the architecture from the user to the layer where the media object is found. In their way, messages are filtered depending on resource availability and local scheduling decisions, normally based on heuristics and a concept of *cost* that keeps the system load balanced.

The use of flexible components, internally based on design patterns, is one of the keys of the architectural flexibility. The scheduling functions of the server is

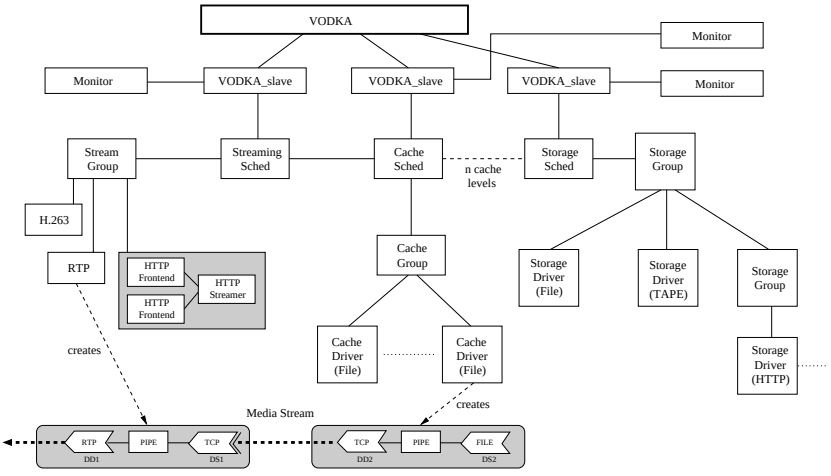


Fig. 1. Processes in a linear configuration of VoDKA

realised in a distributed manner by its processes and components. Every process participating in the distributed scheduling has a function determining local restrictions, given the constraints, the configuration, and the current state of the part of the system that process is aware of. Information about system performance is difficult to obtain due to distributed nature of the VoDKA scheduler, so we try to extract it automatically. The research question is: *how to use formal verification techniques to extract automatically global information from the local restrictions present in the source code and in the system configuration data*.

The Erlang source code of the VoDKA processes involved in the scheduling contains approximately 800 lines of code distributed over nine modules, implementing the kernel part of the video server. Features such as e.g. low level transmission protocols were abstracted away, since they are not needed for performance analysis. Furthermore the release of resources is ignored, meaning that we consider only scenarios where users never release media objects.

4 Extracting Information Using Model Checking

Given a configuration for VoDKA, including its processes, the storage devices, all the restrictions (on bandwidth and number of connections of disk drives, and such), scheduling functions (filtering and admission policies), and cost related functions (resources still available), how can we extract performance information directly from the source code of the system?

For non trivial configurations, this is far from obvious without building the actual configuration and testing it. Answers to such questions, however, are what both the operator of the Video–on–Demand server and the designers of the system are interested in. This problem is rather general. In a design, either concurrent or distributed, where one has many processes that steer a certain functionality, one often finds global properties of the system hidden in several

local properties of the running processes. Here we propose a method to reveal these global properties by a kind of exhaustive simulation of the system.

The proposed methodology consists in using the Erlang source code, and its initial configuration, as the model of the system, and generating its state space. We use the fact that the systems are built using design patterns to abstract from some details of the implementation and keep the size of the state space relatively small. Once we have generated the reduced state space, the performance properties can be extracted from the state graph using model checking.

The properties we analyze are divided into three groups, corresponding to different ways in which the model checking techniques are applied:

Counter-example based capacity properties. Here we search for states with some particular capacity property, and calculate the path with the smallest measure to such a state. For example: "what is the minimum number of movies that can be streamed until a capacity problem occurs?"

Logarithmic search. A fast search algorithm with logarithmic complexity is used to calculate the longest path to a given state. Example: "How many users can watch the movie $m1$ at the same time?"

Scenario-based properties. Ad hoc queries to the model checker. Example: "Is it possible to play $m1$ ten times and $m2$ five times concurrently?"

We are also able to extract architectural information from system messages, to obtain e.g. bottleneck information ("in which system component does the first bottleneck arise?") in order to correctly dimension a system configuration.

In general the method schematically presented here is tool-independent, and has been used successfully to analyse a number of different target systems [10].

5 Results

In this section we discuss how McErlang was used to answer capacity and performance queries regarding VoDKA. To demonstrate the implementation of the method we select a particular property, *first streaming failure due to lack of transfer capacity*. For other examples, see [10].

In the experiments we composed VoDKA with a service user that nondeterministically chose a movie, with a certain preferred bandwidth. The events in the system state space are the messages the particular servers receive and reply to. In our example, the messages are requests for media objects that are passed from one level to the other, and a list of choices propagated in return.

It is clear that we cannot hope to generate the complete state graph for a system configuration with the number of movies that an operational VoD server would have. However, we realized that identifying specific movies are not needed for the performance analysis: it suffices to look at configurations of the system in which we instantiate our storage devices with abstract objects $m1$, $m2$, etc, representing all possible distributions of media in the server. Thus, there is a only one single object that is present both in tape and disk storage; one single object that is present only on tape, one single object that is present only on disk,

etc. The real distribution of movies is a function from media objects to abstract media objects. The user of our tools can therefore still ask the question: "Is it possible to have 30 users watch the movie Star Wars at the same time?". The question is computed for the related abstract object and the answer is valid for all the specific objects of that kind, including the movie the user is asking for. We can as well obtain answers that qualify why the query fails, and a recipe for improvement: "No this is not possible, but if you put one extra copy on CD player 4, then this is possible". Concretely we check whether the property holds for all possible abstract media objects, and if it fails for the desired abstract media objects, find another one for which it succeeds, and compute the difference.

The state spaces, and time of generation using McErlang with different configurations of VoDKA, are summarised below for a particular experiment. For the graph generation, we used the default abstraction (it only orders the processes using their identifiers), and also the default state table, which stores the states in a Erlang ETS (hash) table, and a simple stateless monitor. For each example we indicate the media object storage structure and the number of different media objects available.

Storage Structure	MOs	State space	Time
Tape, CD	3	1207	$<$ 1 second
Tape, CD, Disk	7	28995	61 seconds
Tape, CD, Disk1, 2	15	–	–

Due to the very big size of the state space of the last configuration, its total state space could not be calculated. However, we use this example to demonstrate how the current limitations can be worked around thanks to the flexibility of the McErlang tool and our method for computing capacity which does not require the total state graph to be computed.

We illustrate the method on the problem of computing the shortest path to a fail in the graph (counter-example based property). To extract that information, we needed to write a special monitor for McErlang. The internal state of the monitor was not used, but the monitor was made aware of the minimum counterexample already found during the path search. When given a state to check, the monitor explores the list of processes in the system for that state, and identifies the ones corresponding to users that are about to receive a message from the system saying that the requested media object cannot be streamed. Such a state corresponds to a fail in the graph, and the monitor then calculates a counterexample size. If the new counterexample is shorter than any previously found it is kept, otherwise it is thrown away. In any case, continuing executing from that state is futile as no shorter counter example can be found. In other words, we can prune the total state space in a very effective manner during the search for the shortest path to a fail. This means that, for example, if the smaller counterexample we have already found is 50, we will not explore longer paths.

The percentage of the graph that is now explored depends directly on in which order and with what frequency the counterexamples are found by McErlang. In the table below we show the size of the state spaces generated and the timings

Storage Structure	MOs	State space	Time	Max path	Min path
Tape, CD	3	935	$<$ 1 second	44	44
Tape, CD, Disk	7	5462	16 seconds	176	50
Tape, CD, Disk1, 2	15	25376	93 seconds	309	56

for the improved counterexample search. The column "Max path" is the length of the first counterexample path found, "Min path" the shortest.
Note that in the case of the third configuration, the optimised search has succeeded in locating a counterexample by searching only around 25000 states.

The example above shows the flexibility and power of the McErlang tool. The same approach works fine for verifying other VoDKA capacity properties, including "the longest successful path", "the maximum number of plays" and so on. We have developed tailored monitors that stores the size of the maximum path already found. The flexibility of the McErlang tool can be also used in order to extract bottleneck information or studying the resources needed to add a new component. Being able to look into the whole state of all the processes involved in the system, during the property checking, allows us to provide a more specific diagnosis about the reasons for the bottleneck.

Finally, we have explored the use of McErlang for other kind of properties not directly related to capacity (standard LTL model checking). An example of this is checking if the user always receives an answer after sending a `lookup` message. In order to verify that, we implemented a new monitor where there are two internal states, one *dangerous* state where the user already sent the message and there is an answer message pending to be received, and the state where no answer is pending; the resulting property was verified by McErlang.

6 Conclusions

The proposed approach in this paper deploys formal methods in a rather original way: using model checking tools as general, flexible and efficient graph search algorithms to answer capacity queries about complex distributed systems. Using such techniques we are capable of analysing system configurations with a complexity matching those of VoDKA systems deployed in production.

The method used for calculating the answers to performance queries is related to the method used in [2]. However, in [2] the Erlang source code was first translated into an intermediate process algebra, its state graph constructed by another tool, and finally capacity properties were checked using a standard model checking tool. The key advantages of the McErlang approach are that (i) the model checker supports a much bigger subset of the Erlang language which facilitates model generation, (ii) its model checker is on-the-fly, permitting capacity analysis of systems whose state spaces are too big for complete generation, and (iii) correctness properties can be specified in Erlang, permitting re-use of functions defined already in the source program for data computations. Finally, the use of a modern functional programming language as the basis of the model checker makes the process of tailoring the analysis (e.g., changing abstractions and state table implementation, implementing a new monitor) much quicker.

At the moment our approach is the only one that can automatically verify such global performance properties of a system. Other code level model checkers for Erlang with this accuracy do not exist. For Java, and C, there are similar approaches when it comes to model checking source code [4,8,9]. However, we are not aware of an attempt to use these source code model checkers for analysing a flexible concurrent and distributed architecture in the manner we have accomplished here. In [6] the CADP tool is used for performance evaluation of LOTOS specifications. The authors start from a functionally correct formal specification, and add to it time information related to delays associated with events. In contrast, we use a pure model checking approach, without time information and focusing in the worst case scenario. Also, we extract performance information directly from source code, not from specifications.

Areas for future research include studying to what extent we can improve the process of model construction. That is, can we use real system source code as the input to our methodology, and abstracting automatically from the low level details? Moreover we would like to refine the model of the Video–on–Demand cache, eventually adding the true 'cache' level of the implementation to the verified model. This will produce significant changes in the way properties are extracted from the system (e.g., impacting on the usage of abstract media objects). Finally to improve the user experience, we need to refine the tool diagnostics and interaction. For instance, generating counterexamples more easily understood by the tool users, or making it easier to experiment with different configurations.

References

1. Armstrong, J., Virding, R., Wikström, C., Williams, M.: Concurrent Programming in Erlang. Prentice-Hall, Englewood Cliffs (1996)
2. Arts, T., Sánchez, J.J.: Global scheduler properties derived from local restrictions. In: Proceedings of the ACM Sigplan Erlang Workshop, Pittsburg (2002)
3. Blau, S., Rooth, J.: AXD 301 - a new generation ATM switching system. Ericsson Review 1, 10–17 (1998)
4. Corbett, J., Dwyer, M., Hatcliff, L.: Bandera: A source-level interface for model checking Java programs. In: ICSE 2000, Limerick, Ireland (2000)
5. Fredlund, L., Benac Earle, C.: Model checking Erlang programs: The functional approach. In: ACM Sigplan International Erlang Workshop, Portland, USA (2006)
6. Garavel, H., Hermanns, H.: On combining functional verification and performance evaluation using CADP. In: Eriksson, L.-H., Lindsay, P.A. (eds.) FME 2002. LNCS, vol. 2391, Springer, Heidelberg (2002)
7. Gulías, V., Barreiro, M., Freire, J.L.: VoDKA: Developing a Video-on-Demand server using distributed functional programming. Journal of Functional Programming 15(3), 403–430 (2005)
8. Havelund, K., Pressburger, T.: Model checking JAVA programs using JAVA PathFinder. Software Tools for Technology Transfer 2(4), 366–381 (2000)
9. Holzmann, G.: Design and validation of computer protocols. Prentice-Hall, Englewood Cliffs (1991)
10. Sánchez, J.J.: Penas. From software architecture to formal verification of a distributed system. PhD thesis, Universidade da Coruña, Spain (2006)

Compressing Propositional Proofs by Common Subproof Extraction

Carsten Sinz

Symbolic Computation Group, WSI for Computer Science
University of Tübingen, Germany
carsten.sinz@uni-tuebingen.de

1 Introduction

Propositional logic decision procedures [1,2,3,4,5,6] lie at the heart of many applications in hard- and software verification, artificial intelligence and automatic theorem proving [7,8,9,10,11,12]. They have been used to successfully solve problems of considerable size. In many practical applications, however, it is not sufficient to obtain a yes/no answer from the decision procedure. Either a model, representing a sample solution, or a justification, why the formula possesses none is required. So, e.g. in declarative modeling or product configuration [9,10] an inconsistent specification given by a customer corresponds to an unsatisfiable problem instance. To guide the customer in correcting his specification, a justification why it is erroneous can be of great help. In the context of model checking proofs are used, e.g., for abstraction refinement [11], or approximative image computations through interpolants [13]. In general, proofs are also important for certification through proof checking [14].

Different propositional logic solvers are already available that can generate proofs, e.g., zChaff [4], MiniSAT [6], booleforce [15], or ebddres [16]. Most of them are extensions of DPLL (Davis-Putnam-Logemann-Loveland [1,2]) solvers, which play a dominant role in handling real-world problems. DPLL-based solvers generate *lemmas* or *learned clauses*. The derivations of these lemmas (called *proof chains*), taken together, form a resolution refutation in case the input formula is unsatisfiable. Beame *et al.* [17] have lain the theoretical groundwork for such proofs. But whereas the basic theoretical questions are resolved, practical shortcomings still exist. So the generated proofs can be of considerable size, in the order of hundreds of megabytes for realistic examples. This rises the question whether it is possible to generate smaller proofs. First steps in this direction were undertaken and include techniques like *core extraction* [14], proof generation using other algorithms than DPLL (e.g., BDD-based methods [16]), or computation of minimally unsatisfiable subsets (MUS) [18].

In this paper, we introduce a new technique for compressing proofs that is based on identifying common sub-proofs and merging them. It is intended to be used as a post-processor for proofs generated with one of the methods above. It can therefore also be combined with existing approaches for obtaining short proofs.

Merging is applied on the level of *proof chains*. Common parts of proof chains are united, thereby producing possibly shorter proofs.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 547–555, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Basic Notions

Throughout this paper, we assume propositional logic formulas in conjunctive normal form (CNF), built over a finite set of variables V. A formula F therefore is a conjunction of *clauses*, where a clause is a disjunction of *literals*, and a literal L is either a variable or its negation, i.e. $L \in V \cup \neg V$. We also use *set notation* for formulas in CNF. Assume that a formula $F = C_1 \wedge \cdots \wedge C_m = (L_{1,1} \vee \cdots \vee L_{1,k_1}) \wedge \cdots \wedge (L_{m,1} \vee \cdots \vee L_{m,k_m})$ in CNF is given. As the connectives are uniquely determined by the form, we can equivalently represent F as a set of clauses $\{C_1, \ldots, C_m\}$, where $C_i = \{L_{i,1}, \ldots, L_{i,k_i}\}$. A clause with no literals is called the *empty clause* (denoted by $\square$), a clause with a single literal is called a *unit clause*, its sole literal is called a *unit*. By $\mathbb{B} := \{0, 1\}$ we denote the set of Boolean constants, where we identify 0 with *false* and 1 with *true*. A (partial) *valuation* (or *assignment*) is a function $\alpha : V' \to \mathbb{B}$, where $V' \subseteq V$. If $V' = V$, we say that the valuation is *complete*. A valuation can equivalently be represented as a set of non-contradictory literals, i.e. a set $A \subseteq V \cup \neg V$ with $\{x, \neg x\} \not\subseteq A$ for all variables $x \in V$. The valuation induced by such a set A is defined by $\alpha(x) = 1$ iff $x \in A$, $\alpha(x) = 0$ iff $\neg x \in A$. A valuation given as a set of literals A satisfies a formula F, if $C \cap A \neq \emptyset$ for all clauses $C \in F$. A valuation A immediately contradicts a formula F, if there is a clause $C \in F$ with $\neg L \in A$ for all literals $L \in C$. A formula F is said to be *satisfiable*, if there is a (complete) valuation that satisfies F. Otherwise it is called *unsatisfiable*.

Unit propagation is a simplifying operation on clause sets. For a formula F with unit clauses $U = \{\{u_1\}, \ldots, \{u_r\}\} \subseteq F$, unit propagation removes all clauses containing at least one $u \in U$ from F, and deletes all literals $\neg u$ (with $u \in U$) from clauses in F. We denote the clause set resulting from repeated application of unit propagation to a formula F until a fix point is reached by $\mathrm{UP}(F)$. F is satisfiable iff $\mathrm{UP}(F)$ is. Given a partial valuation A, the result of *unit propagation assuming* A, $\mathrm{UP}_A(F)$, is defined as $\mathrm{UP}_A(F) := \mathrm{UP}(F \cup A^*)$, where $A^* = \{\{a\} \mid a \in A\}$. If, after unit propagation for a clause set F and a valuation A, $\mathrm{UP}_A(F)$ contains the empty clause, assignment A contradicts F. A *conflict graph* can then be generated that captures the individual unit propagation steps that lead to the empty clause (or conflict). The nodes of the conflict graph are literals of unit clauses occurring during unit propagation plus an additional node for the conflict. We identify nodes with literals, and assume an additional "conflict literal" $\emptyset$ for the conflict node. Nodes corresponding to units from the assignment A are called *assumption nodes*, all other nodes, besides the conflict node, are called *propagation nodes*. Moreover, propagation nodes and the conflict node are marked with a clause $C \in F \cup \{\square\}$, and $L \in C$ holds for the literal L of each node besides the conflict node. There is an edge from each literal $\neg L'$ with $L' \in C \setminus \{L\}$ to the literal L, indicating that $\{L\}$ became a new unit clause due to the presence of unit clauses $\{\neg L'\}$ for all other literals L' of C. The conflict node has either two literal nodes $L', \neg L'$ (a complementary literal pair) as predecessors, in which case it is marked with the empty clause, or it is marked with a clause $C \in F$ and has all literals $\neg L'$ for $L' \in C$ as predecessors.

Resolution [19] is a proof system for formulas in CNF, which consists of a single derivation rule which allows to infer a new clause from two given clauses C and D:

$$\frac{C \qquad D}{(C \setminus \{x\}) \cup (D \setminus \{\neg x\})}$$

A sequence of resolution steps, using clauses from F as axioms, is called a *refutation proof* for F, if the last derived clause in the sequence is the empty clause. For each unsatisfiable formula there is a resolution refutation proof witnessing that F is contradictory.

3 Propositional Proof Generation

In this section, we give a short introduction on how proofs are generated in DPLL-style solvers incorporating clause learning. We do not give the traditional presentation of the DPLL algorithm here (which can be found, e.g., in [4]), but reduce it to the facets required to understand how proofs are generated within these solvers.

The DPLL algorithm takes a propositional logic formula, say F, in CNF as input, and tries to find a satisfying assignment $\alpha : V \to \mathbb{B}$ for F (built over variable set V). It does so by extending a partial assignment—starting with the empty assignment—until a contradiction is reached. As soon as this happens, a conflict graph is generated and from this graph a *conflict clause* is derived which captures the reason, why this partial assignment is inadmissible. The conflict clause is then added to F, excluding this partial assignment for the rest of the search. Then the solver backtracks by flipping the value assigned to one of the variables in the conflict clause and continues its search, until all partial assignments have been checked. If no satisfying assignment was found, F is unsatisfiable, and a refutation proof for F may be generated.

3.1 A Proof-Theoretic View of the DPLL Algorithm

As we are only interested in proof generation, we use a slightly abstracted version of the DPLL algorithm in the following, in which the details of the backtracking-procedure are left out. This abstracted algorithm works as follows. We assume an input formula F that does not contain the empty clause (in this case the proof would be trivial):

1. Generate a minimal partial assignment π_i that does not immediately contradict F, but does so after unit propagation assuming π_i. This gives rise to a conflict graph G, which consists of the implications (unit propagations) that led to the conflict. If no such assignment π_i exists, stop (the formula then is satisfiable and no proof can be generated).
2. Add the weakest clause C_i to F, which "forbids" the partial assignment π_i, and generate a proof for C_i (called a *proof chain*) out of the conflict graph G.
3. If π_i is the empty assignment (then C_i is the empty clause $\square$), the refutation for F is complete. Otherwise continue with Step 1.

Before we illustrate the operation of this algorithm on an example, we want to comment on some details: In step 1, if no suitable assignment π_i exists, this may happen for one of the following reasons: (a) all partial assignments immediately contradict F or (b) all partial assignments that do not immediately contradict F, neither do so after unit propagation. Case (a) cannot happen, as the empty assignment never contradicts F, because $\square \notin F$ by assumption (this invariant always holds in step 1, as the algorithm stops

in step 3, as soon as the empty clause is added to F). If case (b) holds, a satisfying assignment can be constructed in the following way: start with the empty assignment and apply unit propagation. By assumption, no contradiction arises. Then add an arbitrary literal to the assignment. This does not produce an immediate conflict, as in each clause there are at least two unassigned (open) literals. Re-apply unit propagation. Again, by assumption, no conflict occurs. By repeating the process of adding literals to the assignment and applying unit propagation, one finally arrives at a complete assignment that satisfies F.

3.2 Conflict Graphs and Proof Chains

We now show in detail how proofs are constructed by the algorithm given in the last section. The construction process uses conflict graphs as intermediate objects, converts them to proof chains, and builds a complete proof out of these proof chains.

Example 1. Let $F = \{\{a, b\}, \{\neg a, b\}, \{\neg a, \neg b, c, d\}, \{\neg b, c, \neg d\}, \{\neg c, e\}, \{\neg c, \neg f\}, \{\neg e, f\}, \{a, c, d\}\}$. Assume that the algorithm first chooses the partial assignment $\pi_1 = \{\neg b\}$. Unit propagation results in the additional units a and $\neg a$, which induces a conflict. The corresponding conflict graph and the resulting proof chain are shown in Fig. 1(i). We explain below how the proof chain is constructed from the conflict graph.

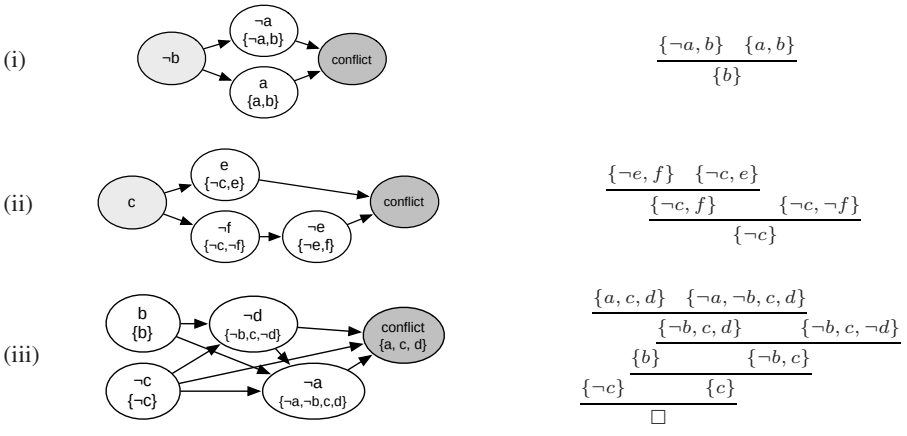


Fig. 1. Conflict graphs and proof chains for Example 1. Assumption nodes are shown in light grey, conflict nodes in dark grey, and propagation nodes in white.

Now, clause $C_1 = \{b\}$ is added to F, thus excluding the partial assignment π_1 for the rest of the search. Proceeding with partial assignments $\pi_2 = \{c\}$ and $\pi_3 = \emptyset$ (shown in Fig. 1(ii) and (iii)), we obtain two further proof chains. The last one derives the empty clause. Taking all proof chains together, we have a resolution refutation proof for F.

Note, that each proof chain has a quite simple structure: each resolution step involves at least one clause from F (thus the resolution proof for a proof chain is *input* and therefore

also *linear*), and each variable is resolved upon at most once (thus proof chains are also *regular*). Beame *et al.* [17] call such proofs *trivial* resolution proofs. Though the proofs for individual proof chains are trivial, this is typically not the case for the complete proof consisting of multiple proof chains.

To construct a proof chain from a conflict graph, clauses of the graph's nodes are resolved in inverse topological order: resolution starts at the conflict node and proceeds back through the graph until the sources of the graph are reached. The proof chain resulting from a given conflict graph is not unique, as there might be different topological orderings of the nodes. Our compression method makes use of this indeterminism by choosing the optimal topological ordering for merging proof chains.

4 Proof Compression Method

The input to our compression method is an already existing proof. We are using *booleforce*[1] to generate such proofs. Booleforce's output consists of a set of *proof chains*. Each proof chain consists of an *identifier* (a natural number), an *assertion* (a clause) and a *derivation* (a set of clauses, given by their identifiers). The semantics of a proof chain is such that by resolving the derivation's clauses (in a not further specified order) the chain's assertion can be proved. Assertions are represented in a DIMACS-like clause format, derivations are space-separated lists of identifiers, ending with 0 as a terminal symbol. For example, the proof line `10 2 3 -4 0 2 5 7 9 0` indicates that by resolving clauses 2, 5, 7, and 9 we obtain clause 10, consisting of the literals 2, 3, and -4. Clauses from the original problem instance are represented in the same way, but possess an empty derivation.

In proofs generated this way, chains that share large parts of their derivations frequently occur. In a proof of the pigeon hole principle (PHP), e.g., the following two proof chains show up, which differ only by three clauses ($6 \leftrightarrow 17, 8 \leftrightarrow 18, 10 \leftrightarrow 19$):

```
20 -15 -31 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 0
21 -15 -32 0 1 2 3 4 5 17 7 18 9 19 11 12 13 14 15 16 0
```

So, there are (in principle) good chances to produce a shorter proof for these two assertions. The general idea to produce shorter proofs is to merge common subproofs occurring in different proof chains, where a subproof consists of a common subset of clauses. By merging them, repetitions of identical subproofs may be avoided. The problem is to identify subproofs that may be factored out. We want to illustrate the problem by an example:

Example 2. Assume two proof chains as shown below.

$$\dfrac{\dfrac{\{x,a\}\quad\{\neg a,b\}}{\{x,b\}}\qquad\{\neg b,c\}}{\{x,c\}} \qquad\qquad \dfrac{\dfrac{\{y,a\}\quad\{\neg a,b\}}{\{y,b\}}\qquad\{\neg b,c\}}{\{y,c\}}$$

Here, it is possible to factor out the common subproof

$$\dfrac{\{\neg a,b\}\quad\{\neg b,c\}}{\{\neg a,c\}}$$

[1] *Booleforce* is available for download at `http://fmv.jku.at/booleforce`.

by re-ordering the given resolution proofs so that the clauses $\{\neg a, b\}$ and $\{\neg b, c\}$ are resolved first. Then the resolvent $\{\neg a, c\}$ is used to prove $\{x, c\}$ using $\{x, a\}$ and $\{y, c\}$ using $\{y, a\}$. Whereas the original proof chains require four resolution steps, the proof with the subproof factored out needs only three.

As a second example, consider the two proof chains

$$\dfrac{\dfrac{\{e, \neg a, \neg c\} \quad \{\neg a, c\}}{\{e, \neg a\}} \qquad \{a, b\}}{\{e, b\}} \qquad \text{and} \qquad \dfrac{\dfrac{\{f, \neg a, \neg c\} \quad \{\neg a, c\}}{\{f, \neg a\}} \qquad \{a, b\}}{\{f, b\}} .$$

Here, it is not possible to factor out a common subproof. A possible candidate for merging would be the subproof

$$\dfrac{\{\neg a, c\} \quad \{a, b\}}{\{b, c\}} ,$$

but when this proof is factored out, the desired clauses may not be re-derived (without introducing additional resolutions), as, e.g., resolving $\{b, c\}$ with $\{e, \neg a, \neg c\}$ produces the weaker clause $\{e, \neg a, b\}$, but not the desired result $\{e, b\}$.

4.1 The Merging Criterion

So we are left with the question, when it is possible to factor out a common subproof of two proof chains. We propose to base this decision not on the proof chains, but on the conflict graphs, as the conflict graphs allow for different resolution orders depending on the topological ordering of the nodes. We therefore define the notation of an *isolatable subgraph* of a conflict graph, which corresponds to a resolution subproof that may safely be factored out. The definition is based on a partial ordering $\succ_G$ on the nodes of a conflict graph, which is defined as follows: $n_1 \succ_G n_2$ iff there is a path from node n_1 to node n_2 in the conflict graph G.

Definition 1. *A subgraph S of a conflict graph G is called an* isolatable subgraph *of G, when all of the following holds:*

1. *If n_1, n_2 are nodes of S with $n_1 \succ_G n_2$, then $n' \in S$ for all $n_1 \succ_G n' \succ_G n_2$.*
2. *There is a unique minimal node m (w.r.t. $\succ_G$) in S.*
3. *m is the only node with edges leaving S (i.e. ending in $G \setminus S$).*

Criteria 1 and 2 ensure that all clauses of S can be resolved. Criterion 3 ascertains that no intermediate resolvent in S is used in other parts of the proof outside of S. We can now formulate our merging criterion:

Lemma 1. *Let G_1, G_2 be conflict graphs and S a common subgraph of G_1 and G_2. The resolution proof corresponding to S can be factored out, if S is an isolatable subgraph of both G_1 and G_2.*

Here, by common subgraphs we understand subgraphs that coincide in structure as well as node labeling (literals and clauses). Lemma 1 is the basis of our merging algorithm, which works as follows:

Input: A refutation proof Π of a formula F produced by a DPLL-style algorithm with clause learning.

1. Select two proof chains γ_1, γ_2 from Π.
2. Compute conflict graphs G_1 and G_2 for γ_1 and γ_2.
3. Determine all common maximal isolatable subgraphs $\{S_1, \ldots, S_k\}$ of G_1 and G_2.
4. Factor out the resolution proofs for $\{S_1, \ldots, S_k\}$ and adapt γ_1 and γ_2 accordingly.
5. Continue with 1., until no further common subgraphs can be extracted.
Output: Compressed proof Π'.

We use a heuristic in Step 1 of this algorithm to find suitable candidates for merging. This heuristics just considers the "neighborhood" of a clause (the n follow-up clauses in the proof) and tries to maximize the number of common clauses in the chains' derivations. It works well, as similar proof chains typically occur proximate to each other.

5 Implementation and Experimental Results

We have implemented the proof compression algorithm of the previous section in a C++ command line tool called *ProofCompress*. *ProofCompress* runs under both Linux and Mac OS X and is available from the author of this paper upon request. Besides compressing proofs, it can also dump conflict graphs in a format suitable for graph layout tools like AT&T's GraphViz (`http://www.graphviz.org`). The graph dump feature also allows to highlight common isolatable subgraphs of two proof chains.

Table 1. Results obtained with *ProofCompress* on various SAT instances

Instance	original #resolutions	#resolutions after 100 compr. rounds		#resolutions after 1000 compr. rounds		runtime 100 rounds	runtime 1000 rounds
aim-200-2_0n4	119	110	(92.4%)	110	(92.4%)	0.03	0.03
dubois20	773	622	(80.5%)	608	(78.7%)	0.24	1.00
dubois30	1,291	1,053	(81.6%)	1,021	(79.1%)	0.39	1.98
dubois50	1,703	1,364	(80.1%)	1,316	(77.3%)	0.61	3.78
een-pico-p01-75	207,274	177,136	(85.5%)	160,852	(77.6%)	1576.39	2517.36
hole6	6,257	5,776	(92.3%)	5,122	(81.9%)	1.21	11.23
hole7	38,475	37,795	(98.2%)	36,094	(93.8%)	7.91	67.38
hole8	223,165	222,198	(99.6%)	220,123	(98.6%)	87.49	449.11
ibm-04-26k25	935	894	(95.6%)	894	(95.6%)	0.84	4.76
ibm-04-6023k100	256,858	196,433	(76.5%)	171,578	(66.8%)	255.80	756.17
longmult3	7,122	4,674	(65.6%)	4,266	(59.9%)	1.84	14.18
longmult4	32,811	22,392	(68.3%)	16,380	(49.9%)	8.86	45.76
longmult5	150,443	130,909	(87.0%)	92,446	(61.5%)	80.47	325.32
manol-pipe-g6bi	736,758	708,803	(96.2%)	584,510	(79.3%)	1567.02	3330.61
mutcb10	134,319	131,492	(97.9%)	121,449	(90.4%)	33.78	227.11
ssa0432-003	679	511	(75.3%)	511	(75.3%)	0.32	0.42

Table 1 displays results obtained with *ProofCompress* on an Intel Core 2 Duo processor running at 2.4 GHz under Linux. The first column shows the instance name (instances are available from `http://fmv.jku.at/sat-race-2006` and `http://www.satlib.org`), the second column the number of resolution steps in a proof generated by *booleforce*. We used this proof as input to *ProofCompress* and let it run for 100 resp. 1000 compression rounds on each instance. Proof size (in number of resolutions) and run time for *ProofCompress* under these two settings are shown in the remaining four columns. We observed proof compression ratios of up to 49.9% in our experiments.

6 Related and Future Work, Conclusion

Other work on obtaining smaller propositional proofs, besides what is mentioned in the introduction, was done by H. Amjad [20]. His proof compression method works on complete refutation proofs, and compresses them by reordering resolution steps. As it works on complete proofs, it may not scale to larger instances as well as our method.

We have presented an algorithm to compress proofs as generated by DPLL-style SAT solvers with clause learning. Our method works by merging similar proof chains. Besides having the potential to compress existing proofs, we suppose that our method may also be built into existing SAT-solvers to directly produce smaller proofs.

References

1. Davis, M., Putnam, H.: A computing procedure for quantification theory. In: JACM 7 (1960)
2. Davis, M., Logemann, G., Loveland, D.: A machine program for theorem-proving. Communications of the ACM 5(7) (1962)
3. Marques-Silva, J.P., Sakallah, K.A.: GRASP — a new search algorithm for satisfiability. In: Proc. ICCAD 1996 (1996)
4. Moskewicz, M., Madigan, C., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an efficient SAT solver. In: Proc. DAC 2001 (2001)
5. Goldberg, E., Novikov, Y.: BerkMin: A fast and robust SAT-solver. In: Proc. DATE 2002 (2002)
6. Eén, N., Sörensson, N.: An extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT 2003. LNCS, vol. 2919, Springer, Heidelberg (2004)
7. Biere, A., Cimatti, A., Clarke, E., Zhu, Y.: Symbolic model checking without BDDs. In: Cleaveland, W.R. (ed.) ETAPS 1999 and TACAS 1999. LNCS, vol. 1579, Springer, Heidelberg (1999)
8. Velev, M., Bryant, R.: Effective use of boolean satisfiability procedures in the formal verification of superscalar and VLIW microprocessors. J. Symb. Comput. 35(2) (2003)
9. Shlyakhter, I., Seater, R., Jackson, D., Sridharan, M., Taghdiri, M.: Debugging overconstrained declarative models using unsatisfiable cores. In: Proc. ASE 2003 (2003)
10. Sinz, C., Kaiser, A., Küchlin, W.: Formal methods for the validation of automotive product configuration data. AI EDAM 17(1) (2003)
11. McMillan, K., Amla, N.: Automatic abstraction without counterexamples. In: Garavel, H., Hatcliff, J. (eds.) ETAPS 2003 and TACAS 2003. LNCS, vol. 2619, Springer, Heidelberg (2003)
12. Xie, Y., Aiken, A.: Scalable error detection using boolean satisfiability. In: Proc. POPL 2005 (2005)
13. McMillan, K.: Interpolation and SAT-based model checking. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, Springer, Heidelberg (2003)
14. Zhang, L., Malik, S.: Validating SAT solvers using an independent resolution-based checker: Practical implementations and other applications. In: Proc. DATE 2003 (2003)
15. Biere, A.: Booleforce (2006), Available at `http://fmv.jku.at/booleforce`
16. Jussila, T., Sinz, C., Biere, A.: Extended resolution proofs for symbolic SAT solving with quantification. In: Biere, A., Gomes, C.P. (eds.) SAT 2006. LNCS, vol. 4121, Springer, Heidelberg (2006)
17. Beame, P., Kautz, H., Sabharwal, A.: Towards understanding and harnessing the potential of clause learning. J. Artif. Intell. Res. (JAIR) 22 (2004)

18. Dershowitz, N., Hanna, Z., Nadel, A.: A scalable algorithm for minimal unsatisfiable core extraction. In: Biere, A., Gomes, C.P. (eds.) SAT 2006. LNCS, vol. 4121, Springer, Heidelberg (2006)
19. Robinson, J.A.: A machine-oriented logic based on the resolution principle. JACM 12 (1965)
20. Amjad, H.: Compressing propositional refuations. In: Proc. AVoCS 2006 (2006)

Verification of ACTL Properties by Bounded Model Checking*

Wenhui Zhang

Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences, Beijing, China
zwh@ios.ac.cn

Abstract. SAT-based bounded model checking has been introduced as a complementary technique to BDD-based symbolic model checking in recent years and a lot of successful work has been done with this approach. The success is mostly due to the efficiency of error-detection. Verification of valid properties depends on a completeness threshold that could be too large to be practical. We discuss an approach to checking valid ACTL (the universal fragment of CTL) properties similar to bounded model checking of ACTL. Bounded model checking of ATCL has been considered in [8]. Given a model M and an ACTL formula φ, a series of k-models of M are constructed for $k = 0, 1, 2, ...$, and the process for checking φ proceeds as follows: start with the 0-model, if the model does not satisfy the negation of φ, use 1-model and so forth, until the negation of φ is satisfied or until a bound of k is reached. A general bound for k is the number of states of M. Trying all k-models up to the bound in order to obtain a conclusion is obviously not desirable. For attacking this problem, we propose an approach to (partly) avoid the use of such a bound.

1 Introduction

With the papers of Biere et. al. [1,2] in 1999, SAT-based bounded model checking of LTL properties has been introduced as a complementary technique to BDD-based symbolic model checking, and a lot of successful work has been done with this approach. The idea has later also been applied to the verification of ACTL (the universal fragment of CTL) properties [8]. The efficiency of this method is based on the observation that if a system is faulty then only a fragment of its state space is sufficient for finding an error. For valid properties, the length (completeness threshold) that needed to be checked in order to certify that the system is error free is usually quite big, such that it is not practical to use this approach for checking systems that are error free with respect to given properties, because for reasonably large systems, this length would possibly be large enough to make the verification intractable. Our research aims at methods for avoiding

* Supported by the National Natural Science Foundation of China under Grant No. 60573012 and 60421001, and the National Grand Fundamental Research 973 Program of China under Grant No. 2002cb312200.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 556–563, 2007.
© Springer-Verlag Berlin Heidelberg 2007

this problem when checking systems that are error free. A work in this direction related to LTL properties has been carried out and presented in [10]. This paper proposes an approach to (partly) avoid the dependence on such a completeness threshold for verification of ACTL properties. The basic idea is to specify an encoding such that if it is unsatisfiable then the encoded problem instance has no witness.

2 Computation Tree Logic

Computation tree logic (CTL) is a propositional branching-time temporal logic [4] introduced by Emerson and Clarke as a specification language for finite state systems. Let AP be a set of proposition symbols. The set of CTL formulas is defined as follows:

- Every member of AP is a CTL formula.
- Logical connectives of CTL include: $\neg$, $\wedge$, $\vee$, and $\rightarrow$.
  If φ and ψ are CTL formulas, then so are $\neg\varphi$, $\varphi \wedge \psi$, $\varphi \vee \psi$, and $\varphi \rightarrow \psi$.
- Temporal operators include:
  EX, EF, EG, EU, ER, AX, AF, AG, AU, and AR.
  If φ and ψ are CTL formulas, then so are:
  $EX\,\varphi$, $EF\,\varphi$, $EG\,\varphi$, $E(\varphi\,U\,\psi)$, $E(\varphi\,R\,\psi)$, $AX\,\varphi$, $AF\,\varphi$, $AG\,\varphi$, $A(\varphi\,U\,\psi)$, and $A(\varphi\,R\,\psi)$.

A model for CTL formulas is a Kripke structure $M = \langle S, T, I, L\rangle$ where S is a set of states, $T \subseteq S \times S$ is a transition relation which is total, $I \subseteq S$ is a set of initial states and $L : S \rightarrow 2^{AP}$ is a labeling function that maps each state of S to a set of propositions that are assumed to be true at that state. A sequence $\pi = \pi_0\pi_1 \cdots$ of S is a path of M, if $T(\pi_i, \pi_{i+1})$ holds for all $i \geq 0$.

Definition 1. *Let M be a model, s a state, p a proposition symbol, φ and ψ CTL formulas. $M, s \models \varphi$ denotes that φ is true at the state s in M. Let π be a path of M. The relation $\models$ is defined as follows:*

$M, s \models p$	*iff* $p \in L(s)$
$M, s \models \neg\varphi$	*iff* $M, s \not\models \varphi$
$M, s \models \varphi \wedge \psi$	*iff* $(M, s \models \varphi)$ *and* $(M, s \models \psi)$
$M, s \models \varphi \vee \psi$	*iff* $(M, s \models \varphi)$ *or* $(M, s \models \psi)$
$M, s \models \varphi \rightarrow \psi$	*iff* $(M, s \models \varphi)$ *implies* $(M, s \models \psi)$
$M, s \models EX\varphi$	*iff* $\exists\pi.(\pi_0 = s \wedge M, \pi_1 \models \varphi)$
$M, s \models EF\varphi$	*iff* $\exists\pi.(\pi_0 = s \wedge \exists k \geq 0.(M, \pi_k \models \varphi))$
$M, s \models EG\varphi$	*iff* $\exists\pi.(\pi_0 = s \wedge \forall k \geq 0.(M, \pi_k \models \varphi))$
$M, s \models E(\varphi U \psi)$	*iff* $\exists\pi.(\pi_0 = s \wedge \exists k \geq 0.(M, \pi_k \models \psi \wedge \forall j < k.(M, \pi_j \models \varphi)))$
$M, s \models E(\varphi R \psi)$	*iff* $\exists\pi.(\pi_0 = s \wedge (\forall j \geq 0.(M, \pi_j \models \psi) \vee$ $\exists k \geq 0.((M, \pi_k \models \varphi) \wedge \forall j \leq k.(M, \pi_j \models \psi))))$
$M, s \models AX\varphi$	*iff* $\forall\pi.(\pi_0 = s \rightarrow M, \pi_1 \models \varphi)$
$M, s \models AF\varphi$	*iff* $\forall\pi.(\pi_0 = s \rightarrow \exists k \geq 0.(M, \pi_k \models \varphi))$
$M, s \models AG\varphi$	*iff* $\forall\pi.(\pi_0 = s \rightarrow \forall k \geq 0.(M, \pi_k \models \varphi))$

$M, s \models A(\varphi U \psi)$ *iff* $\forall \pi.(\pi_0 = s \rightarrow \exists k \geq 0.(M, \pi_k \models \psi \wedge \forall j < k.(M, \pi_j \models \varphi)))$
$M, s \models A(\varphi R \psi)$ *iff* $\forall \pi.(\pi_0 = s \rightarrow (\forall j \geq 0.(M, \pi_j \models \psi) \vee$
$\exists k \geq 0.((M, \pi_k \models \varphi) \wedge \forall j \leq k.(M, \pi_j \models \psi))))$

A CTL formula is in negation normal form (NNF), if the symbol $\rightarrow$ does not appear in the formula and $\neg$ is applied only to proposition symbols. Every formulas can be transformed into an equivalent formula in NNF.

The sublogic ECTL is the subset of CTL formulas that can be transformed into NNF formulas not containing the temporal operators AX, AF, AG, AU, AR. Dually, the sublogic ACTL is the subset of CTL formulas that can be transformed into NNF formulas not containing the temporal operators EX, EF, EG, EU, ER. We only consider formulas in NNF. An ECTL formula φ is true in M, iff φ is true at some initial state of the model M. An ACTL formula φ is true in M, iff φ is true at all initial states of the model M.

For simplicity, we fix the model under consideration to be $M = \langle S, T, I, L \rangle$, and in the sequel, M refers to this model, unless otherwise is stated. Let $k \geq 0$. A k-path of M is a path $\pi = s_0 \cdots s_k$ where $(s_i, s_{i+1}) \in T$ for $i = 0, ..., k-1$. The k-model for M is a structure $M_k = \langle S, P_k, I, L \rangle$ where P_k is the set of all different k-paths of M. Let $\pi = \pi_0 \cdots \pi_k \in P_k$. Let $loop(\pi)$ denote $\bigvee_{i=0}^{k} T(\pi_k, \pi_i)$.

Definition 2. *Let M_k be a k-model, s a state, p a proposition symbol, φ and ψ ECTL formulas. $M_k, s \models_k \varphi$ denotes that φ is true at the state s in M_k. Let $[i]$ denote the set $\{0, ..., i\}$. The relation $\models_k$ is defined as follows:*

$M_k, s \models_k p$	*iff* $p \in L(s)$
$M_k, s \models_k \neg p$	*iff* $p \notin L(s)$
$M_k, s \models_k \varphi \wedge \psi$	*iff* $(M_k, s \models_k \varphi)$ *and* $(M_k, s \models_k \psi)$
$M_k, s \models_k \varphi \vee \psi$	*iff* $(M_k, s \models_k \varphi)$ *or* $(M_k, s \models_k \psi)$
$M_k, s \models_k EX\varphi$	*iff* $\exists \pi.(\pi_0 = s \wedge M_k, \pi_1 \models_k \varphi)$
$M_k, s \models_k EF\varphi$	*iff* $\exists \pi.(\pi_0 = s \wedge \exists i \in [k].(M_k, \pi_i \models_k \varphi))$
$M_k, s \models_k EG\varphi$	*iff* $\exists \pi.(\pi_0 = s \wedge loop(\pi) \wedge \forall i \in [k].(M_k, \pi_i \models_k \varphi))$
$M_k, s \models_k E(\varphi U \psi)$	*iff* $\exists \pi.\pi_0 = s \wedge$ $\exists i \in [k].(M_k, \pi_i \models_k \psi \wedge \forall j \in [i-1].(M_k, \pi_j \models_k \varphi))$
$M_k, s \models_k E(\varphi R \psi)$	*iff* $\exists \pi.(\pi_0 = s \wedge ((loop(\pi) \wedge \forall j \in [k].(M_k, \pi_j \models_k \psi)) \vee$ $\exists i \in [k].((M_k, \pi_i \models_k \varphi) \wedge \forall j \in [i].(M_k, \pi_j \models_k \psi))))$

An ECTL formula φ is true in the k-model M_k, denoted $M_k \models_k \varphi$, iff φ is true at some initial state of the model M_k. According to [8], the following holds.

Theorem 1. *Let φ be an ECTL formula. Then the following hold: i) $M \models \varphi$ iff there is a $k \geq 0$ such that $M_k \models_k \varphi$. ii) If $M_k \models_k \varphi$, then $M_{k+1} \models_{k+1} \varphi$.*

3 Encoding of ECTL Formulas

Since every formula has an equivalent formula in NNF and the two temporal operators EF and ER can be defined by EU and EG as follows: $EF\varphi = E(trueU\varphi)$ and $E(\varphi R \psi) = E(\psi U(\varphi \wedge \psi)) \vee EG\psi$, we may only consider formulas of the form

$\varphi \vee \psi, \varphi \wedge \psi, EX\varphi, EG\varphi, E(\varphi U \psi)$ constructed recursively from propositions and the negation of propositions.

Let m be the number of states of M. A state is to be encoded by a truth assignment to $n = \lceil \log_2(m) \rceil$ propositional variables. Let u be a state variable. Then u has n parts $u^0, ..., u^{n-1}$. Let $k \geq 0$. Let N_k be the number of different k-paths of M (or one may use an over-approximation and let N_k be m^{k+1}). Let $u_{i,0}, ..., u_{i,k}$ be a finite sequence of state variables on some path with each $i \in \{1, ..., N_k\}$. The approximated transition relation for a given k is encoded as follows.

$$[[M]]_k := \bigwedge_{i=1}^{N_k} \bigwedge_{j=0}^{k-1} T(u_{i,j}, u_{i,j+1})$$

Let $p \in AP$ be a proposition symbol and $p(u)$ represent the propositional formula representing the states in which p is true according to L. For the encoding of ECTL formulas, we have the following definition.

Definition 3. *Let $k \geq 0$. Given a state u and a formula φ. The encoding $[[\varphi, u]]_k$ is defined as follows.*

$[[p, u]]_k$	$= p(u)$
$[[\neg p, u]]_k$	$= \neg p(u)$
$[[\varphi \vee \psi, u]]_k$	$= [[\varphi, u]]_k \vee [[\psi, u]]_k$
$[[\varphi \wedge \psi, u]]_k$	$= [[\varphi, u]]_k \wedge [[\psi, u]]_k$
$[[EX\varphi, u]]_k$	$= \bigvee_{i=1}^{N_k} (u = u_{i,0} \wedge [[\varphi, u_{i,1}]]_k)$
$[[EG\varphi, u]]_k$	$= \bigvee_{i=1}^{N_k} (u = u_{i,0} \wedge \bigwedge_{j=0}^{k} [[\varphi, u_{i,j}]]_k)$
$[[E(\varphi U \psi), u]]_k$	$= \bigvee_{i=1}^{N_k} (u = u_{i,0} \wedge$ $(\bigvee_{j=0}^{k} ([[\psi, u_{i,j}]]_k \wedge \bigwedge_{t=0}^{j-1} [[\varphi, u_{i,t}]]_k) \vee \bigwedge_{t=0}^{k} [[\varphi, u_{i,t}]]_k))$

where $[[\varphi, u_{i,j}]]_k$ denotes true if $j > k$.

While in the encoding in [8], the basic idea is that if the encoding is satisfiable then the encoded problem instance has a witness, on the other hand, this encoding has the property that if the encoding is unsatisfiable then the encoded problem instance has no witness. This is to be elaborated in the sequel.

Let $I(u)$ be the propositional formula such that an assignment to u satisfies $I(u)$ if and only if the assignment of u represents an initial states of M. Based on the definitions of $I(u)$, $[[M]]_k$ and $[[\varphi, u]]_k$, we define the encoding $[[M, \varphi]]_k$ for satisfiability testing.

Definition 4. *Let u be a state variable and φ be an ECTL formula. $[[M, \varphi]]_k$ is defined as follows.* $[[M, \varphi]]_k := I(u) \wedge [[M]]_k \wedge [[\varphi, u]]_k$.

The three components in the definition are explained as follows: $I(u)$ restricts the assignment to u such that it represents an initial state of M, $[[M]]_k$ restricts the assignment to $u_{i,j}$ such that it represents valid k-paths of M, and $[[\varphi, u]]_k$ restricts the assignment to $u_{i,j}$ such that the set of k-paths represents a potential

witness (that has potential to be extended to a witness) of φ starting from u in M. We have the following lemma.

Lemma 1. *Let φ be an ECTL formula, and $k \geq 0$. Then the following holds. i) If $M_k \models_k \varphi$, then $[[M, \varphi]]_k$ is satisfiable. ii) If $[[M, \varphi]]_{k+1}$ is satisfiable, then $[[M, \varphi]]_k$ is satisfiable.*

The two properties in this lemma can be proved by structural induction on φ. Details of the proof are omitted. Following this lemma and Theorem 1, we obtain the following theorem.

Theorem 2. *Let φ be an ECTL formula. If $M \models \varphi$, then $[[M, \varphi]]_k$ is satisfiable for all $k \geq 0$.*

A reformulation gives that if $[[M, \neg\psi]]_k$ is unsatisfiable for some $k \geq 0$, then $M \models \psi$ for an ACTL formula ψ. This makes the basis for SAT-based verification of ACTL formulas in the style of bounded model checking in our approach, which is to check the unsatisfiability of the encoding $[[M, \neg\psi]]_k$ for $k = 0, 1, 2, ...$

4 Number of Paths

One of the particular aspects of ECTL is that we have to deal with a set of bounded paths instead of a single path (in the case of LTL). The number of paths (i.e. N_k) considered in the previous section is not a practical one for carrying out a verification. For practical application, the sufficient number of paths involved in a verification depends on k and the formula to be verified, and this number can be calculated based on the structure of the formula as follows.

$f_k(p) = f_k(\neg p)$	$= 0 \ if \ p \in AP$
$f_k(\varphi \wedge \psi)$	$= f_k(\varphi) + f_k(\psi)$
$f_k(\varphi \vee \psi)$	$= max(f_k(\varphi), f_k(\psi))$
$f_k(EX\varphi)$	$= f_k(\varphi) + 1$
$f_k(EG\varphi)$	$= (k+1) \cdot f_k(\varphi) + 1$
$f_k(E(\varphi U \psi))$	$= k \cdot f_k(\varphi) + max(f_k(\varphi), f_k(\psi)) + 1$

For proving that there is no substructure of the model can satisfy φ, it is sufficient to involve $f_k(\varphi)$ k-paths. Let $b \geq 1$. We first define the restricted transition relation in which the number of paths is restricted to b as follows.

Let $k \geq 0$.

$$[[M]]_k^b := \bigwedge_{i=1}^{b} \bigwedge_{j=0}^{k-1} T(u_{i,j}, u_{i,j+1})$$

Then we define the encoding of ECTL formulas with the number of involved paths restricted to b.

Definition 5 (Restricted Translation of ECTL Formulas). *Let $k \geq 0$. Given a state u and a formula φ. The encoding $[[\varphi, u]]_k^b$ is defined as follows.*

$[[p,u]]_k^b$	$= p(u)$
$[[\neg p,u]]_k^b$	$= \neg p(u)$
$[[\varphi \vee \psi,u]]_k^b$	$= [[\varphi,u]]_k^b \vee [[\psi,u]]_k^b$
$[[\varphi \wedge \psi,u]]_k^b$	$= [[\varphi,u]]_k^b \wedge [[\psi,u]]_k^b$
$[[EX\varphi,u]]_k^b$	$= \bigvee_{i=1}^{b}(u = u_{i,0} \wedge [[\varphi,u_{i,1}]]_k^b)$
$[[EG\varphi,u]]_k^b$	$= \bigvee_{i=1}^{b}(u = u_{i,0} \wedge \bigwedge_{j=0}^{k}[[\varphi,u_{i,j}]]_k^b)$
$[[E(\varphi U\psi),u]]_k^b$	$= \bigvee_{i=1}^{b}(u = u_{i,0} \wedge$
	$(\bigvee_{j=0}^{k}([[\psi,u_{i,j}]]_k^b \wedge \bigwedge_{t=0}^{j-1}[[\varphi,u_{i,t}]]_k^b) \vee \bigwedge_{t=0}^{k}[[\varphi,u_{i,t}]]_k^b))$

where $[[\varphi,u_{i,j}]]_k^b$ *denotes true if* $j > k$.

Similarly, we define the encoding $[[M,\varphi]]_k^b$ for satisfiability testing.

Definition 6. *Let* u *be a state variable and* φ *be an ECTL formula.* $[[M,\varphi]]_k^b$ *is defined as follows.* $[[M,\varphi]]_k^b := I(u) \wedge [[M]]_k^b \wedge [[\varphi,u]]_k^b$.

According to the definition, $[[M,\varphi]]_k^{N_k}$ is the same as $[[M,\varphi]]_k$.

Lemma 2. *Let* φ *be an ECTL formula.* $[[M,\varphi]]_k$ *is satisfiable iff* $[[M,\varphi]]_k^{f(k(\varphi)}$ *is satisfiable.*

This lemma can be proved by structural induction on φ. Details of the proof are omitted. Following this lemma and Theorem 2, we obtain the following theorem.

Theorem 3. *Let* φ *be an ECTL formula. If* $M \models \varphi$, *then* $[[M,\varphi]]_k^{f_k(\varphi)}$ *is satisfiable for all* $k \geq 0$.

5 Verification Approach

Theorem 3 provides a basis for the verification of valid ACTL properties. For minimizing the number of propositions used in the SAT formulas, we base the verification on the following corollary where the variable u (implicitly) in Theorem 3 is replaced by $u_{1,0}$ which is already in $[[M]]_k^{f_k(\varphi)}$.

Let $[[M,\varphi]]_k^* := I(u_{1,0}) \wedge [[M]]_k^{f_k(\varphi)} \wedge [[\varphi,u_{1,0}]]_k^{f_k(\varphi)}$. The satisfiability of $[[M,\varphi]]_k^*$ is the same as that of $[[M,\varphi]]_k^{f_k(\varphi)}$. We have the following corollary of Theorem 3.

Corollary 1. *Let* φ *be an ECTL formula. If* $M \models \varphi$, *Then* $[[M,\varphi]]_k^*$ *is satisfiable for all* $k \geq 0$.

For the verification of an ACTL property ψ, we check the unsatisfiability of the encoding $[[M,\neg\psi]]_k^*$ for $k = 0, 1, 2, ...$ until the unsatisfiability is shown or the bound of resource is reached. In the former case, we have a conclusion that ψ is valid in M. In the latter case, we do not have a conclusion.

Note that this is an incomplete approach. We may combine different approaches, such as induction or interpolation based approaches [9,7,3,5], for achieving advantages of each of the approaches. For the positive application of this approach, we

have carried out a sequence of case studies that illustrate the potential benefit of the approach and how the verification may be done.

We use a sequence of cases to illustrate our approach. The number of processes in a case ranges from 2 to 7. Let $p_0, ..., p_{n-2}, q, r$ be variables of the domain $\{0, 1\}$ and $\oplus$ be the function: addition modulo 2. Let the system be consist of n processes. A, B and C_i for $i = 0, ..., n-3$ (each is a sequential process executed in parallel to each other with the interleaving semantics) with the following specification:

$A: \ r = r \oplus 1; p_0 = p_0 \oplus 1$
$B: \ p_{n-2} = p_{n-2} \oplus 1; q = q \oplus 1$
$C_i : p_i = p_i \oplus 1; p_{i+1} = p_{i+1} \oplus 1; q = q \oplus 1$

We consider two property, one with the temporal operator AR and the other with AU. Let n be the number of processes. Let
$\varphi(n) = A((p_0 \vee \cdots \vee p_{n-2})Rq)$, and
$\zeta(n) = AXA((p_0 \vee \cdots \vee p_{n-2})U(\neg q \vee \neg r))$.
We check whether the system satisfies the property $\varphi(n)$ and $\zeta(n)$. The experimental results (for minimal k that makes the formula unsatisfiable) are presented in Table 1.

Table 1. Experimental data for 3, 4, 5, 6, 7 processes

Property	k	Variables	Clauses	Time	SAT
$\varphi(3)$	2	50	360	0.0	no
$\varphi(4)$	2	65	625	0.0	no
$\varphi(5)$	2	80	962	0.0	no
$\varphi(6)$	2	95	1371	0.0	no
$\varphi(7)$	2	110	1852	0.0	no

Property	k	Variables	Clauses	Time	SAT
$\zeta(3)$	3	143	1079	0.0	no
$\zeta(4)$	5	286	3075	0.1	no
$\zeta(5)$	7	477	6611	0.6	no
$\zeta(6)$	9	716	12119	8.5	no
$\zeta(7)$	11	1003	20031	191.8	no

The time (in seconds) is that used by zChaff [6] for satisfiability checking. The table shows that the efficiency of verification may depend on whether we need large k for verification. Sometimes, a small k is sufficient for the verification of certain properties, such as φ in this example. For verifying φ for $n = 100$ processes, the verification time by zChaff is 8.4 seconds with 1505 propositional variables and 361297 clauses in the CNF formula. The approach as presented in [8] would not be as efficient for verification in such cases, since the diameter of the model is relatively large (with a huge number of reachable states), and the number k has a huge impact (possibly exponential, as indicated in the verification of $\zeta(k)$) on the verification time.

6 Concluding Remarks

Bounded model checking based on SAT has been introduced in 1999 as an complementary technique to BDD-based symbolic model checking. The basic idea is to search for a counter example of a particular length (as short as possible) and to

generate a propositional formula that is satisfied iff such a counter example exists. One difficulty with this approach for verifying valid properties is that it depends on a completeness threshold which could be very large. For attacking this problem, this work presents an encoding of pairs of model and formula in SAT for the purpose of verification of valid ACTL properties. The illustrative example has shown that it is useful for checking formulas that are valid in a model, in the sense that the iteration may stop (as shown in the example) before the completeness threshold (with respect to the original approach) is reached. Although the model presented in the example is simple with the completeness threshold bounded by a relatively small number, it is easy to construct models by extending the model, such that the completeness threshold is larger than any given number, while the verification can stop when k reaches a relatively small number for some given properties. Therefore the benefit of the use of this approach could be very large compared to the use of completeness threshold, and this extends the practical capability of SAT based model checking to the verification of valid ACTL properties. The approach presented in this work is an incomplete approach which can be combined with the original approach presented in [8], so that it is possible to know whether an ACTL property holds when a completeness threshold is reached, and at the same time we have a possibility to detect error or report validity before such a completeness threshold is reached.

References

1. Biere, A., Cimmatti, A., Clarke, E.M., Zhu, Y.: Symbolic Model Checking without BDDs. In: Cleaveland, W.R. (ed.) ETAPS 1999 and TACAS 1999. LNCS, vol. 1579, pp. 193–207. Springer, Heidelberg (1999)
2. Biere, A., Cimatti, A., Clarke, E.M., Fujita, M., Zhu, Y.: Symbolic Model Checking Using SAT Procedures instead of BDDs
3. Das, S., Dill, D.L.: Successive Approximation of Abstract Transition Relations. LICS 2001 51–60. DAC 317–320 (1999)
4. Allen Emerson, E., Clarke, E.M.: Using Branching-time Temporal Logics to Synthesize Synchronization Skeletons. Science of Computer Programming 2(3), 241–266 (1982)
5. Jhala, R., McMillan, K.L.: Interpolation and SAT-Based Model Checking. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, pp. 1–13. Springer, Heidelberg (2003)
6. Moskewicz, M.W., Madigan, C.F., Zhao, Y., Zhang, L., Malik, S.: Chaff: Engineering an Efficient SAT Solver. In: DAC (2001)
7. de Moura, L., Ruess, H., Sorea, M.: Bounded Model Checking and Induction: From Refutation to Verification. In: Hunt Jr., W.A., Somenzi, F. (eds.) CAV 2003. LNCS, vol. 2725, pp. 14–26. Springer, Heidelberg (2003)
8. Penczek, W., Wozna, B., Zbrzezny, A.: Bounded Model Checking for the Universal Fragment of CTL. Fundamenta Informaticae 51, 135–156 (2002)
9. Sheeran, M., Singh, S., Stlmarck, G.: Checking Safety Properties Using Induction and a SAT-Solver. In: Johnson, S.D., Hunt, Jr., W.A. (eds.) FMCAD 2000. LNCS, vol. 1954, pp. 108–125. Springer, Heidelberg (2000)
10. Zhang, W.: SAT-Based Verification of LTL Formulas. In: Brim, L., Haverkort, B., Leucker, M., van de Pol, J. (eds.) FMICS 2006 and PDMC 2006. LNCS, vol. 4346, Springer, Heidelberg (2007)

Application of Linear Hybrid Cellular Automata to Stream Ciphers*

A. Fúster-Sabater[1], P. Caballero-Gil[2], and M.E. Pazo-Robles[1]

[1] Instituto de Física Aplicada, C.S.I.C., Serrano 144, 28006 Madrid, Spain
amparo@iec.csic.es, eugenia.pazo@iec.csic.es
[2] DEIOC, University of La Laguna, 38271 La Laguna, Tenerife, Spain
pcaballe@ull.es

Abstract. Binary sequences generated by a class of linear cellular automata (the so-called multiplicative polynomial cellular automata) can be written as solutions of linear difference equations. At the same time, such solutions equal the output sequences of well known LFSR-based keystream generators. Cryptographic properties of the sequences generated by multiplicative polynomial cellular automata have been analyzed. In addition, an algorithm to model nonlinear LFSR-based generators in terms of these linear automata has been developed.

1 Introduction

Stream ciphers have extensive applications in secure communications, e.g. wireless systems, due to practical advantages such as easy implementation, high speed and good reliability. When designing a stream cipher, the main goal is to expand a short key into a long pseudorandom keystream sequence in such a way that it should not be possible to reconstruct the short key from the keystream. In this work, we focus our attention on binary keystream sequence generators based on Linear Feedback Shift Registers (LFSRs) [7] whose output sequences are combined in a nonlinear way. Such generators produce keystreams with high linear complexity, long period and good statistical properties, see [1], [4] and [11].

On the other hand, one-dimensional binary Cellular Automata (CA) [2] have been found to generate exactly the same Pseudo Noise sequences (PN-sequences) as those of the LFSRs [5]. Nevertheless, in contrast with the LFSRs, the binary sequences generated by adjacent cells in CA are not correlated. In this sense, linear CA are better sequence generators than common LFSRs.

At any rate, the great advantage of the above mentioned CA is that multiple generators designed in terms of LFSRs as nonlinear structures preserve the linearity when they are expressed under the form of CA. The question that arises in a natural way is: are there one-dimensional linear CA able to produce the sequence obtained from any LFSR-based nonlinear generator? The answer is yes and, in fact, this paper tackles the problem of given a particular

* Work supported by Ministerio de Educación y Ciencia (Spain) Projects SEG2004-02418 and SEG2004-04352-C04-03.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 564–571, 2007.
© Springer-Verlag Berlin Heidelberg 2007

LFSR-based generator how to find one-dimensional CA that reproduce exactly its output sequence. More precisely, in this work it is shown that wide classes of LFSR-based nonlinear generators, Clock-Controlled generators, Cascade-Clock-Controlled generators, Shrinking generator or the generators producing Kasami sequences, GMW sequences, No sequences, Klapper sequences ... (see [12]), can be modelled in terms of linear CA. All the previous sequences are included in the class of *interleaved sequences* [8], that is pseudorandom sequences satisfying a common property: each sequence can be decomposed into a collection of shifts of an unique PN-sequence. This kind of sequences can be obtained from linear multiplicative polynomial CA. That is CA made out of a basic structure concatenated a number of times, see [5]. Therefore, the goal of this work is twofold:

- To analyze linear multiplicative polynomial CA with emphasis on the cryptographic parameters of their generated sequences (period, linear complexity, characteristic polynomial, number of different output sequences etc.).
- To model the above nonlinear LFSR-based generators in terms of linear multiplicative polynomial CA.

Once the generators have been linearized, all the theoretical background on linear CA found in the literature can be applied to their analysis and/or cryptanalysis.

2 Fundamentals of Cellular Automata

One-dimensional cellular automata can be described as L-cell registers [9], whose cell contents are updated at the same time instant according to a particular k-variable function (the *transition rule*) denoted by Φ. If the function Φ is a linear function, so is the cellular automaton. Moreover, if $k = 2r + 1$, then the content of the *i-th* cell at time $n + 1$ depends on the contents of k neighbor cells at time n in the following way:

$$x_{n+1}^i = \Phi(x_n^{i-r}, \ldots, x_n^i, \ldots, x_n^{i+r}) \ \ (i = 1, ..., L). \tag{1}$$

The number of cells L is the length of the automaton. CA are called *uniform* whether all cells evolve under the same rule while CA are called *hybrid* whether different cells evolve under different rules. If permanent null contents are supposed adjacent to the extreme cells, then the automata are called *null*. The transition rules we are going to consider in this paper are described as follows :

$$\begin{array}{cc} \text{Rule 90} & \text{Rule 150} \\ x_{n+1}^i = x_n^{i-1} \oplus x_n^{i+1} & x_{n+1}^i = x_n^{i-1} \oplus x_n^i \oplus x_n^{i+1} \end{array}$$

Indeed, the content of the *i-th* cell at time $n + 1$ depends on the contents of either two different cells (rule 90) or three different cells (rule 150) at time n. The symbol $\oplus$ denotes addition modulo 2 among cell contents. Remark that both transition rules are linear. This work deals exclusively with one-dimensional linear null hybrid CA with rules 90 and 150.

For the above mentioned rules, the different states of the automaton are grouped in closed cycles. The number of different output sequences for a particular cycle is $\leq L$ as the same sequence (although shifted) may appear simultaneously in different cells. At the same time, all the sequences in a cycle will have the same period and linear complexity [9]. Moreover, any of the output sequence of the automaton can be produced at any cell provided that the right state cycle is chosen. Next, additional definitions are introduced. A Multiplicative Polynomial Cellular Automaton is defined as a cellular automaton whose characteristic polynomial is of the form $P_M(X) = (P(X))^p$ where p is a positive integer. If $P(X)$ is a primitive polynomial [7], then the automaton is called a Primitive Multiplicative Polynomial Cellular Automaton.

A widely accepted measure of the unpredictability of a sequence is its *linear complexity* [10] that can be defined as follows: The linear complexity of a sequence is the shortest linear recursion satisfied by such a sequence.

For a binary sequence $\{x_n\}$, the linear recurrence relationship that specifies its *n-th* element can be written:

$$x_n + \sum_{j=1}^{r} c_j \; x_{n-j} = 0, \qquad n \geq r \tag{2}$$

where the sequence elements (x_n) as well as the coefficients (c_j) belong to $GF(2)$. The linear recursion can be expressed as a linear difference equation:

$$(E^r + \sum_{j=1}^{r} c_j \; E^{r-j}) \; x_n = 0, \qquad n \geq 0 \tag{3}$$

where E is the shifting operator that operates on x_n, i.e. $Ex_n = x_{n+1}$. The characteristic polynomial of the difference equation (3) is:

$$P(X) = X^r + \sum_{j=1}^{r} c_j \; X^{r-j}. \tag{4}$$

Let $P(X)$ be a primitive polynomial of degree r and $\alpha \in GF(2^r)$ one of its roots. In this case [10],

$$\alpha, \; \alpha^2, \; \alpha^{2^2}, \ldots, \; \alpha^{2^{(r-1)}} \tag{5}$$

are the r roots of such a polynomial. If $P_M(X) = (P(X))^p$, then the roots of $P_M(X)$ will be the same as those of $P(X)$ but with multiplicity p. Consequently, all the binary sequences generated by a primitive multiplicative polynomial cellular automaton will satisfy the linear difference equation:

$$(E^r + \sum_{j=1}^{r} c_j \; E^{r-j})^p \; x_n = 0, \qquad n \geq 0. \tag{6}$$

Our analysis focuses on all the possible solutions $\{x_n\}$ of this equation.

3 Cryptographic Properties of Multiplicative Polynomial Cellular Automata

In this section, the characteristics of the sequences obtained from this kind of CA are considered in terms of their cryptographic parameters: period, linear complexity and number of different sequences.

3.1 Period of the Generated Sequences

The generic solutions of the equation (6) are linear combinations of $p \cdot r$ solutions of the form:

$$x_n = \sum_{i=0}^{p-1} \left(\binom{n}{i} \sum_{j=0}^{r-1} A_i^{2^j} \ \alpha^{2^j n} \right), \tag{7}$$

where $S_n^i = \sum_{j=0}^{r-1} A_i^{2^j} \ \alpha^{2^j n}$ represents the *n-th* element of a PN-sequence [10] of period $2^r - 1$ and $A_i \in GF(2^r)$. Thus, the binary sequence $\{x_n\}$ can be written as the sum of p times the same PN-sequence starting at different elements given by A_i and weighted by a binomial coefficient

$$\{x_n\} = \sum_{i=0}^{p-1} \binom{n}{i} \{S_n^i\}. \tag{8}$$

In addition, each binomial coefficient defines a succession of binary values with a constant period p_i. Therefore, the sequence $\{x_n\}$ is the sum of p sequences of distinct periods $T_i = p_i \cdot (2^r - 1)$ and the period of the total sequence will be:

$$T = \max\{T_i \ (i = 0, \ldots, p-1) \ / \ A_i \neq 0\}. \tag{9}$$

It can be noticed that the period of the different sequences $\{x_n\}$ generated by a multiplicative polynomial cellular automaton depends on the choice of the coefficients A_i in the equation (7). All the sequence generated at the same state cycle have the same period. Nevertheless, the same automaton can produce sequences with distinct periods depending on the state cycle considered.

3.2 Linear Complexity of the Generated Sequences

According to [12], the linear complexity of a sequence equals the number and multiplicity of characteristic polynomial roots that appears in the linear recurrence relationship. Therefore, coming back to the equation (7) the linear complexity of $\{x_n\}$ can be computed. In fact, we have r roots each of them with multiplicity p. Thus, if i_{max} is the greatest value of i $(i = 0, \ldots, p-1)$ for which $A_i \neq 0$, then the linear complexity LC of the sequence $\{x_n\}$ will be:

$$LC = (i_{max} + 1) \cdot r \ . \tag{10}$$

It can be noticed that the linear complexity of the different sequences $\{x_n\}$ generated by a multiplicative polynomial cellular automaton depends on the choice of the coefficients A_i in (7). All the sequence generated at the same state cycle have the same linear complexity. Nevertheless, the same automaton can produce sequences with distinct linear complexities depending on the state cycle considered.

3.3 The Number of Different Generated Sequences

In order to count the number of different sequences $\{x_n\}$ generated, the choice of the coefficients A_i in the equation (7) must be considered. Different situations can take place:

- If $A_i = 0 \ \ \forall\, i$, then all the cells of the cellular automaton will generate the identically null sequence.
- If $A_0 \neq 0$ and $A_i = 0 \ \ \forall\, i > 0$, then all the cells of the cellular automaton will generate a unique PN-sequence $\{S_n^0\}$ of period $T_0 = 2^r - 1$ and characteristic polynomial $P(X)$.
- If $A_0 \in GF(2^r), A_1 \neq 0$ and $A_i = 0 \ \ \forall\, i > 1$, then there will be $2^r \cdot (2^r - 1)$ possible choices of (A_0, A_1). According to subsection (3.1), the period of these sequences will be $T_1 = p_1 \cdot (2^r - 1)$. Thus, the number of different sequences for these values of A_i is:

$$N_1 = \frac{2^r \cdot (2^r - 1)}{p_1 \cdot (2^r - 1)} = 2^{r-1}. \tag{11}$$

- In general, if $A_0, A_1, \ldots, A_{i-1} \in GF(2^r),\ A_i \neq 0$ and $A_j = 0 \ \ \forall\, j > i$, then there will be $2^{(i\,r)} \cdot (2^r - 1)$ possible choices of $(A_0, A_1, \ldots, A_i)$. According to subsection (3.1), the period of these sequences will be $T_i = p_i \cdot (2^r - 1)$. Thus, the number of different sequences for these values of A_i is:

$$N_i = \frac{2^{(i\,r)} \cdot (2^r - 1)}{T_i} = \frac{2^{(i\,r)}}{p_i}. \tag{12}$$

The total number of different sequences obtained from a multiplicative polynomial cellular automaton will be:

$$N_{total} = \sum_{i=0}^{p-1} N_i. \tag{13}$$

In this computation the null sequence is excluded.

4 Linear Modelling by Concatenation of Primitive Cellular Automata

In the previous section, structural properties of the sequences obtained from multiplicative polynomial cellular automata have been considered. Now the particular form of these automata is analyzed. In order to simplify the notation, CA

can be represented by means of binary strings with the following codification: 0 = rule 90 and 1 = rule 150. Since the characteristic polynomial of these automata is $P_M(X) = (P(X))^p$, it seems quite natural to construct a multiplicative polynomial cellular automaton by concatenating p times the basic automaton (or the reverse version). In this way, the construction of a linear model based on CA is carried out by the generic algorithm:
Input: The parameters of a nonlinear keystream generator.

- *Step 1:* Determine the irreducible factor $P(X)$ of the characteristic polynomial of each interleaved sequence.
- *Step 2:* Compute the pair of basic CA whose characteristic polynomial is the irreducible factor $P(X)$.
- *Step 3:* For each one of these basic CA, construct by successive concatenations a cellular automaton able to generate the original interleaved sequence.

Output: Two linear CA producing the corresponding keystream sequence.

The Cattell and Muzio synthesis algorithm [2] presents a method of obtaining two CA (based on rules 90 and 150) corresponding to a given polynomial. Such an algorithm takes as input an irreducible polynomial $Q(X) \in GF(2)[X]$ and computes two linear reversal CA whose output sequences have $Q(X)$ as characteristic polynomial. The total number of operations required for this algorithm is linear in the degree of the polynomial and is listed in [2](Table II, page 334). The method is efficient for all cryptographic applications.

4.1 Linearization of the Shrinking Generator

The shrinking generator [3] is a typical example of cryptographic generator with characteristic polynomial $P_M(X)$. The characteristics of this generator can be summarized as follows: The generator is made of two LFSRs, SR1 and SR2, with lengths $L_j (j = 1, 2)$ and characteristic polynomials $P_j(X) (j = 1, 2)$, respectively. The decimation rule is: the bit produced by SR2 is discarded if the corresponding bit of SR1 equals 0. The decimated sequence is just the output sequence of the generator. The generator characteristic polynomial is of the form $P_M(X) = (P(X))^p$, $P(X)$ being a primitive polynomial of degree $r = L_2$ and $2^{(L_1-2)} < p \leq 2^{(L_1-1)}$. Moreover, $P(X)$ is the characteristic polynomial of the cyclotomic *coset* E in $GF(2^{L_2})$ with $E = 2^0 + 2^1 + \ldots + 2^{L_1-1}$, see [7].

The output sequence will be a solution of a linear difference equation corresponding to primitive multiplicative polynomial CA. Consequently, the shrinking generator can be expressed in terms of a lineal model based on CA. A simple example illustrates the modelling procedure.

Input: A shrinking generator characterized by two LFSRs of lengths $L_1 = 3$, $L_2 = 5$ respectively and characteristic polynomial $P_2(X) = X^5+X^4+X^2+X+1$. Now, $p = 2^{L_1-1} = 4$ and $r = L_2 = 5$.

Step 1: $P(X)$ the irreducible factor of the generator characteristic polynomial is:

$$P(X) = X^5 + X^2 + 1 \; .$$

Step 2: The pair of basic CA whose characteristic polynomial is $P(X)$ are:

$$\begin{array}{c} 0\,1\,1\,1\,1 \\ 1\,1\,1\,1\,0 \end{array}$$

Step 3: Computation of the required pair of CA by successive concatenations. For the first automaton:

$$\begin{array}{l} 0\,1\,1\,1\,1 \\ 0\,1\,1\,1\,0\,0\,1\,1\,1\,0 \\ 0\,1\,1\,1\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,1\,1\,1\,0 \text{ (final automaton)} \end{array}$$

For the second automaton:

$$\begin{array}{l} 1\,1\,1\,1\,0 \\ 1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ 1\,1\,1\,1\,1\,1\,1\,1\,1\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1 \text{ (final automaton)} \end{array}$$

For each automaton, the procedure in *Step 3* has been carried out $L_1 - 1$ times. In fact, each basic automaton with complementations has been concatenated $p = 2^{L_1-1}$ times.

Output: Two binary strings codifying the required CA.

In this way, we have obtained a pair of linear CA able to produce the shrunken sequence corresponding to the given shrinking generator.

4.2 Linearization of the Cascade Gollmann Generator

This generator [6] is made out of m cascaded LFSRs with primitive characteristic polynomials of degree n. In fact, the i-th register is controlled by all the previous registers j with $j < i$ in a stop-and-go way. The period of the generated sequence is $T = (2^n - 1)^m$ and the linear complexity is lower bounded by $LC > n(2^n - 1)^{m-1}$. For a very simple example of this generator with $m = 2$ LFSRs with primitive characteristic polynomials of degree $n = 3$, the output sequence is depicted in Table 1. Indeed, the generated sequence of period $T = 49$ can be read

Table 1. Horizontally: Output sequence of a cascade Gollmann generator. In vertical, different shifts of a PN-sequence or its complementary sequence.

$\overline{PN}$	PN	$\overline{PN}$	PN	PN	PN	$\overline{PN}$
1	1	0	0	0	0	1
1	1	0	1	1	1	0
0	0	1	1	1	1	0
1	0	1	1	1	1	0
0	1	0	0	0	0	1
0	0	1	0	0	0	1
0	1	0	1	1	1	0

horizontally. In addition, every vertical column corresponds to shifts of a PN-sequence of characteristic polynomial $P(X) = X^3 + X^2 + 1$ or its complementary sequence. In order to model this kind of generator, the basic automaton to be concatenated is made out of the automaton associated to $P(X)$ plus an additional cell due to the complementarity. The rest of the modelling procedure is analogous to this one described in the previous example.

5 Conclusion

In this work, structural properties of the primitive multiplicative polynomial CA have been analyzed. It is shown that wide classes of LFSR-based sequence generators with cryptographic application can be described in terms of CA-based structures. In this way, sequence generators conceived and designed as complex nonlinear models can be written in terms of simple linear models. The algorithm to convert a given LFSR-based generator into a CA-based model is simple and can be applied to generators in a range of practical interest. The linearity of these cellular models can be advantageously used in the analysis and/or cryptanalysis of such keystream generators.

References

1. Caballero-Gil, P., Fúster-Sabater, A.: A Wide Family of Nonlinear Filter Functions with a Large Linear Span. Information Sciences 164(4), 197–207 (2004)
2. Cattell, K., Muzio, J.C.: Synthesis of One-Dimensional Linear Hybrid Cellular Automata. IEEE Trans. Computers-Aided Design 15(3), 325–335 (1996)
3. Coppersmith, D., Krawczyk, H., Mansour, Y.: The Shrinking Generator. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 22–39. Springer, Heidelberg (1994)
4. Fúster-Sabater, A.: Run Distribution in Nonlinear Binary Generators. Applied Mathematics Letters 17(12), 1427–1432 (2004)
5. Fúster-Sabater, A., Caballero-Gil, P.: Concatenated Automata in Cryptanalysis of Stream Ciphers. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 611–616. Springer, Heidelberg (2006)
6. Gollmann, D., Chambers, W.: Generators for Sequences with Near-Maximal Linear Equivalence. IEE Proceedings 135, 67–69 (1988)
7. Golomb, S.W.: Shift Register-Sequences. Aegean Park Press, Laguna Hill (1982)
8. Gong, G.: Theory and Applications of q-ary Interleaved Sequences. IEEE Trans on Information Theory 41(2), 400–411 (1995)
9. Kari, J.: Theory of Cellular Automata: a survey. Theoretical Computer Science 334(3), 3–33 (2005)
10. Key, E.L.: An Analysis of the Structure and Complexity of Nonlinear Binary Sequence Generators. IEEE Trans. Informat. Theory 22(6), 732–736 (1976)
11. Pichler, F.: A Highly Nonlinear Cellular FSM-Combiner for Stream Ciphers. In: EUROCAST 2007, Extended Abstracts. IUCTC Universidad de Las Palmas de Gran Canaria, pp. 205–207 (2007)
12. Rueppel, R.A.: Stream Ciphers. In: Simmons, G.J. (ed.) Contemporary Cryptology, The Science of Information, pp. 65–134. IEEE Press, Los Alamitos (1992)

A Highly Nonlinear Cellular FSM-Combiner for Stream Ciphers

Franz Pichler

Systems Theory,Johannes Kepler University Linz
Altenbergerstrasse 69, A-4040 Linz, Austria

Abstract. Combiner construction is an important problem in the design of random generators for applications in stream ciphering. Most constructions use the concept of static combiners based on boolean functions. In this paper we propose the construction of a dynamic combiner which is based on a cellular array of finite state machines which realize a switching network for a XOR combiner with variable length.

1 Requirements for Cryptological Strong Pseudo Random Generators

The realization of a stream cipher device requires a specific kind of pseudo random generator which fulfils the following four properties

- (P1) the generation of the pseudo random stream has to be fast such that high data rates (e.g. 10- 100 MB/sec.) are achieved.
- (P2) the generated stream is statistical close to a pure random stream. This is proved by the passing of a selected battery of statistical tests.
- (P3) when modelling the pseudo random generator by a finite state machine then the initial state is an essential part of the cryptographic key. Its computation from a parts of the generated sequence has to be a provable difficult task.
- (P4) for a pair of identical copies of the pseudo random generator there must exist an effective algorithm to synchronize their cryptographic keys.

Let us give some comments for (P1) to (P4): (P1) has the effect that the generator requires the use of fast hardware for its realization. This means that the concept is based on fast algorithms which can be effectively implemented by current available hardware modules. (P2) requires that the generated sequences are “close” to pure random sequences. The “closeness” is determined by the passing of appropriate statistical tests. Failing of a test indicates the possibility that a deterministic component in the behaviour of the generator can be discovered. Since any test sequence has to be necessarily of finite length there is always the chance that a test fails. Even pure random sequences are candidates of failing in this situation. The required property (P3) is for cryptologic reasons the most important one. It states that that the identification of the cryptographic key has to be provable a computational hard problem. If we assume that the pseudo random generator is modelled by a finite state machine this means that machine identification (the determination of the next

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 572 – 580, 2007.
© Springer-Verlag Berlin Heidelberg 2007

state function and the output function) and state identification has to be computational difficult, regardless of the amount of data which are available to the opponent. In the cryptologic praxis the consideration of a finite state model alone is not sufficient. It concerns only the mathematical functional level in modelling a real existing pseudo random generator. Additional models on the engineering microelectronic level of description have to be taken into account. Property (P4) which asks for the fulfilment of the "synchronisation requirement" depends strongly on the kind of applied technology. There exist very often different methods for its solution. In the case of the use of very fast hardware the concept that both generators are equipped with an exact clock the method "synchronization by state reset" can be used.

2 Architecting a Pseudo Random Generator

It was already stated that the fulfilment of the property (P3), which guarantees that identification of the generator is difficult is most important. In the praxis of stream cipher design the following approach in determining the architecture of the generator has proven to be successful: In the first step pseudo random generators PRG(1). PRG(2), … ,PRG(m) are designed, which meet the properties (P1),(P2) and (P4) but not necessarily (P3). In the second step a "combiner" C is designed, which computes from the sequences $x_1,x_2, \dots ,x_m$ of PRG(1),PRG(2), … ,PRG(m) a resulting sequence $y = C(x_1,x_2, \dots ,x_m)$ such that the overall pseudo random generator prg consisting of the aggregate of PRG(1),PRG(2), … ,PRG(m) in parallel in series with the combiner C meets also the property (P3).

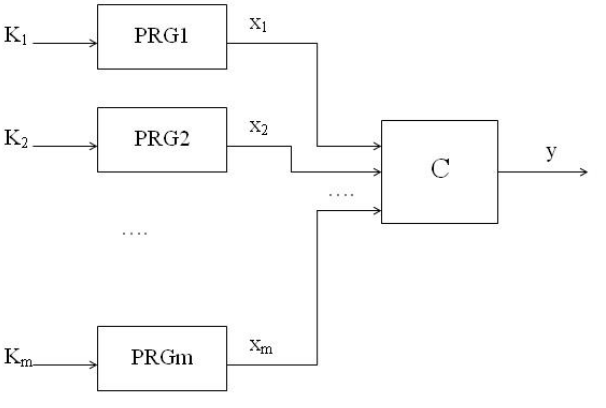


Fig. 1. Architecture of a pseudo random generator

3 Combiner Design

For the design of combiners C different methods have been investigated in the past. In the case that the sequences $x_1,x_2, \dots ,x_m$ which are generated by the different pseudo

random generators are binary data streams the combiner can be defined by a Boolean function $C:B^m \rightarrow B$, where B denotes the set $B=\{0,1\}$. Several authors have contributed to the design of boolean combiners C of such kind ([1],[2],[3]). As an extension of such "static combiners" C we can consider "dynamic combiners" C(t) which varies its function by the clock time t of the generator. A combiner of this kind has been investigated in an earlier work of the author, where for the modeling of the combiner a finite state machine of type "finite memory machine" has been used. In this case the design of a dynamic combiner can be reduced to the design of a static combiner [4]. In the following we describe the design of a dynamic combiner which is modeled by finite state machine which is realized by an array of identical finite state machines.

4 Cellular FSM Combiners

In general we want to investigate the construction of a dynamic combiner C(t) which is modelled by a finite state machine FSM as shown in Figure 2.

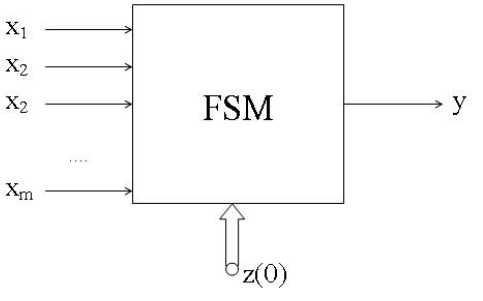


Fig. 2. Block-diagram of a FSM combiner

The different input ports of the FSM combiner receives binary data streams $x_1,x_2, \ldots, x_m$ generated by the associated pseudo random generators PRG(1),PRG(2), … PRG(m). Starting from an initial state z(0) the FSM combiner computes the binary output stream y. It is required that the FSM combiner function is for most of the possible initial states z(0) as considered from the cryptological standpoint satisfying. Then z(0) can be used as an additional key parameter, the FSM combiner key.

A formal description of the FSM can be given as usual by $FSM=(B^m,B,Q,\delta,\lambda)$ where $\delta:QxB^m \rightarrow Q$ and $\lambda:QxB^m \rightarrow B$ denote the next state function and the output function, respectively. For every time point t=0,1,2, … the FSM computes from the input values $x_1(t),x_2(t), \ldots ,x_m(t)$ and the state z(t) the output value of the FSM combiner $y(t) = \lambda(z(t),x_1(t),x_2(t), \ldots ,x_m(t))$.

In our approach to construct the FSM combiner we use an array of m identical finite state machines fsm(1),fsm(2), … , fsm(m) as shown in Fig. 3.

The input streams $x_1,x_2, \ldots ,x_m$ enter the automata array of the FSM combiner level by level by an XOR operation. This motivates us to call the architecture of the FSM combiner as shown in Fig. 3 an "XOR cascade" of finite state machines. Next we

want to investigate the cryptological quality of the FSM combiner. At first we have to determine the kind of finite state machine fsm we use for constructing the XOR cascade. We consider the following types:

(1) fsm is a clock controlled maximal periodic binary linear feedback shift register CCMLFSR
(2) fsm is a clock controlled binary baker register machine CCBRM ([5],[6])
(3) fsm is a clock controlled binary cyclic linear feedback shift register CCCSR.

The investigation of the cryptologic quality of the FSM combiner using the different fsm (1)-(3) can be done by testing a simulation model and by mathematical analysis

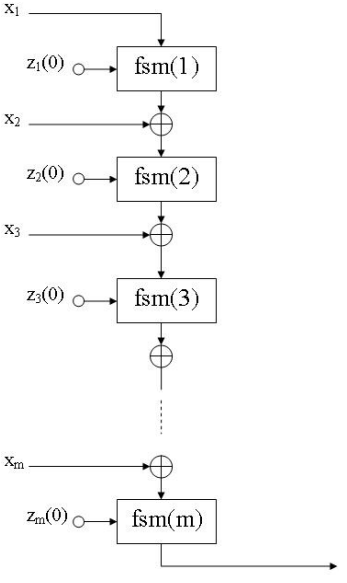


Fig. 3. Realization of the FSM combiner by an array of m finite state machines fsm

5 Cryptanalysis of the FSM Combiner by Simulation

Simulation experiments have been performed for all three types (1)-(3) of finite state machines fsm. Registers of length 512 were chosen such that the key space of the combiner was big enough. The inputs to the combiner were generated by MLFSR´s beeing also of length 512. As test system served “CryptoBench 2006”, a method base system for cryptanalysis as developed by Jochinger [6]. CryptoBench 2006 compares with other existing test systems which are public available.The following tests have been applied:

T1 computation of the linear complexity profile
T2 mono-bit test

T3 two-bit test
T4 Walsh- spectral test

All test showed satisfying results [7].

6 Mathematical Analysis of the FSM Combiner

Mathematical analysis of the FSM combiner is done here only for the case (3) of a clock controlled cyclic linear feedback shift register CCCSR, as shown in Fig. 4 .

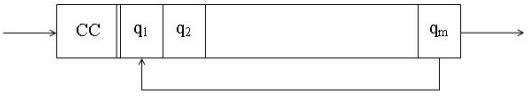


Fig. 4. Block diagram of a clock controlled cyclic linear feedback shiftregister CCCSR

We state the result of the analysis already in advance. It is shown that the FSM combiner C can be expressed in algebraic form by the following term

$$C(x_1,x_2, \dots ,x_m) = p(0)\oplus p(1)x_1\oplus p(2)x_2\oplus \dots \oplus p(m)x_m \quad (1)$$

where $x_1,x_2, .. ,x_m$ denotes the input streams and p(0),p(1), … ,p(m) are booelean polynomials in the state variables of different degree relating to the state machines fsm(1),fsm(2), … ,fsm(m) of the FSM array.We prove this result in the following:

Each CCCSR of the combiner can be represented by a finite state machine CCCSR = (B,B,B^n,δ, β) where the next state function δ is given by $\delta(q,a) := q$ for $a = 0$ and $\delta(q,a) := q\rightarrow 1$ for $a = 1$. Here $q\rightarrow 1$ denotes the cyclic permutation of $q = (q_1,q_2, \dots ,q_n)$ which is computed by the shift operation $\rightarrow 1(q) = (q_n,q_1, \dots ,q_{n-1})$. By usual boolean algebraic notation δ can also be expressed by $\delta(q,a)= q \oplus qa \oplus (q\rightarrow 1)a$. The output function β of CCCSR is defined by $\beta(q_1,q_2, \dots ,q_n) := q_n$.

For each of the different finite state machines fsm(i), i = 1,2, … ,m, we can express the next state function δ and the output function β by

$$\delta(q_{i1},q_{i2}, \dots ,q_{in},a_i) = (q_{i1},q_{i2}, \dots ,q_{in}) \oplus (q_{i1},q_{i2}, \dots ,q_{in})a_i \oplus (q_{in},q_{i1}, \dots ,q_{in-1})a_i \quad (2)$$

and

$$\beta(q_{i1},q_{i2}, \dots ,q_{in}) = q_{in} \quad (3)$$

We compute now for any time point $t > 0$ the output $b_m := y(t)$ of the last finite state machine fsm(m) of the array. We reach this stepwise, starting from the output $b_1 := y_1(t)$ of the first machine fsm(1) of the cascade. We have

$b_1 = q_{1,n} \oplus q_{1n}a_1 \oplus q_{1,n-1}a_1 = q_{1,n} \oplus (q_{1,n} \oplus q_{1,n-1})a_1$ or

$b_1 = P_{10} \oplus P_{11}s_1$ where $P_{10} := q_{1,n}$ and $P_{11} := q_{1,n} \oplus q_{1,n-1}$ with $s_1 := a_1 = x_1(t)$

The output $b_2 := y_2(t)$ of fsm(2) is given by

$b_2 = q_{2,n} \oplus q_{2,n}a_2 \oplus q_{2,n-1}a_2$ or since $a_2 = b_1 \oplus s_2$, with $s_2 := x_2(t)$ we have after ordering with regard to s_1 and s_2

$$b_2 = q_{2,n} \oplus (q_{2,n} \oplus q_{2,n-1})P_{10} \oplus (q_{2,n} \oplus q_{2,n-1})P_{11}s_1 \oplus (q_{2,n} \oplus q_{2,n-1})s_2$$

Introducing the polynomials P_{20},P_{21} and P_{22} by

$P_{20} := q_{2,n} \oplus (q_{2,n} \oplus q_{2,n-1})P_{10}$
$P_{21} := (q_{2,n} \oplus q_{2,n-1})P_{11}$
$P_{22} := (q_{2,n} \oplus q_{2,n-1})$

We are able to write

$$b_2 = P_{20} \oplus P_{21}s_1 \oplus P_{22}s_2$$

By mathematical induction we get after m steps for the output $b_m = y(t)$ of fsm(m) the final result

$$b_m = P_{m0} \oplus P_{m1}s_1 \oplus P_{m2}s_2 \oplus \ldots \oplus P_{mm}s_m \tag{4}$$

where P_{m0},P_{m1}, … ,P_{mm} are boolean polynomials of the state variables of fsm(1),fsm(2), … , fsm(m) given in the following form:

$P_{mm} = q_{m,n} \oplus q_{m,n-1}$
$P_{mm-1} = P_{mm}(q_{m-1,n} \oplus q_{m-1,n-1})$
$P_{mm-2} = P_{mm-1}(q_{m-2,n} \oplus q_{m-2,n-1})$
.
.
$P_{m1} = P_{m2}(q_{1,n} + q_{1,n-1})$

and

$$P_{m0} = q_{m,n} \oplus P_{mm}q_{m-1,n} \oplus P_{mm-1}q_{m-2,n} \oplus \ldots \oplus P_{m2}q_{1,n}$$

To get a more convenient notation we introduce the two-variable linear polynomials K_i by
$K_i := (q_{i,n} \oplus q_{i,n-1})$ for i=1,2, … ,m. Then we have

$P_{mm} = K_m$
$P_{mm-1} = K_mK_{m-1}$
.
.
$P_{m1} = K_mK_{m-1} \ldots K_1$

and

$$P_{mo} = q_{m,n} \oplus K_m q_{m-1,n} \oplus K_m K_{m-1} q_{m-2,n} \oplus ... \oplus K_m K_{m-1} ... K_2 q_{1,n}$$

The polynomials P_{mi} (i=1,2, ... ,m) are, as it easy to see, of degree m-i+1. P_{mm} is of lowest degree 1 and P_{m1} is of highest degree m. The polynomial P_{m0} which appears in the sum of (4) does not effect the cryptological quality of the FSM combiner.

7 Practical Implications

We investigate now the operation of the cellular FSM combiner as given as the result of our mathematical analysis by the formula (4). It will turn out that the combiner realizes in essential a XOR combiner with clockwise varying number of XOR operations. To show this in detail, we introduce the state variables z_i for i=1,2, … ,m by $z_{2i} := q_{i,n}$ and $z_{2i-1} := q_{i,n-1}$. Since $K_i = z_{2i} + z_{2i-1}$ the variables $z_1, z_2, \dots, z_{2m}$ are at the same time the variables of the polynomials $P_{m0}, P_{m1}, \dots, P_{mm}$.

In the case that a polynomial K_i (i=1,2, … ,m) has the value 0 then all of the polynomials P_{mj} with $j \leq i$ have also the value 0. Then the ouput b_m of the combiner is given by

$$b_m = P_m{}^{i+1}{}_0 \oplus P_{mi+1} s_{i+1} \oplus \dots \oplus P_{mm} s_m \tag{5}$$

where $P_m{}^{i+1}{}_0$ denotes the expression

$$P_m{}^{i+1}{}_0 = q_{m,n} \oplus P_{mm} q_{m-1,n} \oplus P_{mm-1} q_{m-2,n} \oplus \dots \oplus P_{mm-i+1} q_{m-i,n} \tag{6}$$

If $K_m = 0$ then (5) reduces to $b_m = q_{m,n}$ and if all K_i (i =1,2, … ,m) are different from zero then the output b_m is given by (4) using the full length of the expression. It is obvious that the situation $K_m=0$ has to be avoided. This can be achieved by selecting the initial state of fsm(m) such that subsequent values 0,0 and 1,1 of the state variables are avoided. The reduction of the size of the key space of the combiner, which follows can be neglected if the length n of the register of fsm(m) is appropriately chosen. By the same method in selecting the initial states of fsm(m),fsm(m-1), … , fsm(m-k+1), k=1,2, … ,m, it can be guaranted that the term $x_{m-k+1} \oplus x_{m-k+2} \oplus \dots \oplus x_m$ is contained in the expression (4) which represents b_m.

The values $q_{1,n}, q_{2,n}, \dots, q_{m-1,n}$ as they appear in the polynomial P_{mo} are the output values of the different finite state machine fsm(1),fsm(2), …, fsm(m). In the formula (4) they have the effect that they are added to the input streams $s_2, s_3, \dots, s_m$ respectively. By the pseudorandom character of $s_2, s_3, \dots s_m$ this does not change the cryptologic quality of these data streams. Finally also the addition of $q_{m,n}$ in formula (4) has no negative effect on $b_m = y(t)$.

Fig. 5 illustrates by a circuit diagram the construction of the computation of the output stream y of the cellular FSM combiner by the application of the formular (4) for each value $b_m = y(t)$.

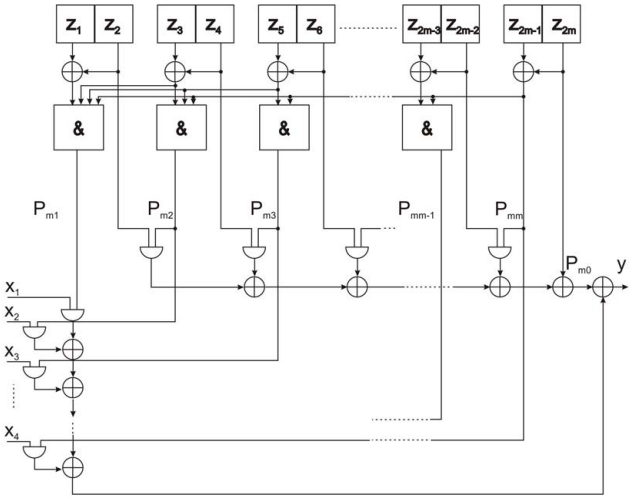


Fig. 5. Circuit diagram for the computation of $b_m = y(t)$

8 Conclusion

The design of the combiner of a pseudo random generator is a crucial part of stream cipher construction. In the past, the design of static combiners which are realized by a boolean function have been discussed in the literature. However, mathematical details were often considered as trade secrets. The design of dynamic combiners have so far not received strong attention. The paper introduced a new type of dynamic combiner which uses an array of specific finite state machines.

For the cases that the array is realized by clock controlled maximal periodic linear feedback shiftregisters CCMLFSR, by clock controlled baker register machines CCBRM or by clock controlled cyclic linear feedback shift registers CCCSR dynamic combiners of this kind showed good results in testing. In the case of CCCSR the mathematical analysis shows, that such a dynamic combiner realizes the XOR-sum of variable length of the different input streams

$x_1, x_2, \ldots, x_m$.

The use of a cellular array of finite state machines for the design of a FSM combiner was in our approach mainly motivated by heuristic reasons. The result shows that a generalisation to other types of coupled FSM`s is possible. Any finite state machine which generates a control for random switching the XOR combiner in its length would be a candidate for a solution.

References

1. Rueppel, R.: Analysis and Design of Stream Ciphers. Springer, Berlin (1986)
2. Meier, W., Staffelbach, O.: Nonlinearity criteria for cryptographic functions. In: Quisquater, J.-J., Vandewalle, J. (eds.) EUROCRYPT 1989. LNCS, vol. 434, pp. 549–562. Springer, Heidelberg (1990)

3. Cobas, G., David, J., Brugos, A.L.: Complexity-Theoretical Approaches to Design and Analysis of Cryptographical Booelean Functions. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 337–345. Springer, Heidelberg (2005)
4. Pichler, F.: Konstruktion korrelationsimmuner Schaltfunktionen und Schaltwerke mittels Walsh-Fourieranalyse. In: Pilz, G. (ed.) Contributions to General Algebra 6 Verlag B.G. Teubner, Stuttgart, Wien, pp. 213–222 (1988)
5. Jochinger, D., Pichler, F.: A New Pseudo-Random Generator Based on Gollmann Cascades of Baker-Register-Machines. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 311–316. Springer, Heidelberg (2005)
6. Jochinger, D.: Ein Pseudo- Zufallsgenerator für die Kryptographie mittels EXOR – Kaskaden von Bäcker- Register- Maschinen. PhD Thesis, Johannes Kepler University Linz (2006)
7. Jochinger, D.: Implementation of a FSM Combiner and Testing with CryptoBench 2006. Internal Report, Kukla Electronics Bad Ischl-Linz (August 2006)

Variations on Neighborhoods in CA

Thomas Worsch[1,⋆] and Hidenosuke Nishio[2]

[1] University of Karlsruhe, Germany
worsch@ira.uka.de
[2] ex. Kyoto University, Japan
yra05762@nifty.com

Abstract. We show how an arbitrary finite number of CA with different local rules can be simulated by CA using *the same local rule* by just changing the (shape of the) neighborhood. In that way one can even achieve universality.

1 Introduction

Usually investigations of cellular automata without further discussion assume some standard neighborhood because it is "without loss of generality". In general this is correct, except, of course, when one is interested in questions specifically concerning neighborhoods.

The rest of this paper is organized as follows: In Section 2 we introduce some basic notation we are going to use later on. The following two sections are devoted to simulations of some CA $\mathcal{A}_i$ by other CA, where the latter only differ in their neighborhoods. Sections 3 presents a solution to the conceptually simpler task of simulating a finite number of CA $\mathcal{A}_i$ which all have the same neighborhood but different local functions by CA $\mathcal{B}_i$ which all have the same local function but different neighborhoods. In Section 4 it will be shown that one can even achieve universality in the following sense: There is *one* local rule which is used by different CA with different neighborhoods in such a way that any CA $\mathcal{A}_i$ with state set $\{0, 1\}$ for its cells can be simulated, even (and in particular) if the initial configuration does not contain any information about the CA to be simulated. That does only depend on the specific neighborhood of the simulating CA.

Throughout the paper we prefer solutions that can be described and used easily over solutions which have optimized running time or set of states.

2 Basics

We will describe the construction in Section 3 for d-dimensional CA and denote by $R = \mathbf{Z}^d$ the set of all cells (in Section 4 we will restrict ourselves to the case $d = 1$). Let Q denote the finite set of states for each cell and N the finite neighborhood containing $n = |N|$ relative offsets to cells. Without loss of generality we assume

⋆ Corresponding author.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 581–588, 2007.
© Springer-Verlag Berlin Heidelberg 2007

that $\mathbf{0} = (0, \ldots, 0) \in N$. Let $\nu : \{0, \ldots, n-1\} \to N$ be a bijection satisfying $\nu(0) = \mathbf{0}$. This function is introduced in order to have a numbering of the neighbors. It allows to use "the same local rule" with different neighborhoods. Equivalently one can think of ν as a vector $(\nu(0), \ldots, \nu(n-1))$.

Global configurations are formalized as mappings $c : R \to Q$; thus $c(\mathbf{j})$ is the state of cell $\mathbf{j}$ in configuration c. The local rule $f : Q^n \to Q$ induces the global function in the usual way. If the CA is $\mathcal{A} = (R, Q, N, \nu, f)$ we write $\mathcal{A} : Q^R \to Q^R$ for the global function.

$$\mathcal{A}(c)(\mathbf{j}) = f(c(\mathbf{j} + \nu(0)), c(\mathbf{j} + \nu(1)), \ldots, c(\mathbf{j} + \nu(n-1))) \; .$$

3 Simulating a Few CA Using One Local Function

For $0 \le i < m$ let $\mathcal{A}_i = (R, Q_A, N_A, \nu_A, f_i)$ denote m cellular automata. Configurations $c : R \to Q_A$ are simply called $\mathcal{A}$-configurations since all these CA have the same set of configurations.

We will describe m CA $\mathcal{B}_i = (R, Q_B, N_i, \nu_i, f_B)$ such that each $\mathcal{B}_i$ simulates $\mathcal{A}_i$. Configurations $c : R \to Q_B$ are simply called $\mathcal{B}$-configurations. Note that *the CA $\mathcal{B}_i$ use the same set of states and the same local function and they only differ in (the shapes of) their neighborhoods.* And the same embedding of $\mathcal{A}$-configurations into $\mathcal{B}$-configurations will be used for all simulations.

We use $Q_B = Q_A \times \{0, 1, \ldots, m-1\}$. For any $q \in Q_B$ we write $s(q)$ and $p(q)$ for the first and second component of q respectively.

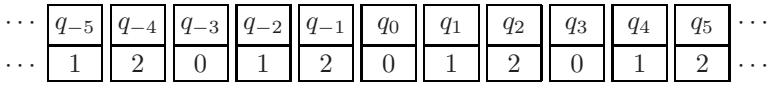
The following embedding $E : Q_A^R \to Q_B^R$ of configurations will be used:

$$E(c)(\mathbf{j}) = (c(\mathbf{j}), \mathbf{j}[0] \bmod m)$$

where $\mathbf{j}[0]$ means the first component of the vector $\mathbf{j}$. For example, if we want to simulate $m = 3$ one-dimensional CA $\mathcal{A}_i$, then $\mathcal{A}$-configurations like



will be embedded into



Lemma 1. *Let c_A be an arbitrary $\mathcal{A}$-configuration and $c_B = E(c_A)$. For all $\mathbf{i}, \mathbf{j} \in R$ and all offsets $\mathbf{x} \in \mathbf{N} \times \{0\}^{d-1}$ holds:*

$$p(c_B(\mathbf{i} + \mathbf{x})) - p(c_B(\mathbf{i})) = p(c_B(\mathbf{j} + \mathbf{x})) - p(c_B(\mathbf{j})) \pmod m$$

This should be obvious since the equation is equivalent to

$$p(c_B(\mathbf{i} + \mathbf{x})) - p(c_B(\mathbf{j} + \mathbf{x})) = p(c_B(\mathbf{i})) - p(c_B(\mathbf{j})) \pmod m \; .$$

Definition 2. *The CA $\mathcal{B}_i$ are specified as follows:*

- $N_i = N \cup \{\mathbf{r_i}\}$ *where* $\mathbf{r_i} = (r_i, 0, \ldots, 0)$ *and* r_i *is the smallest positive integer not occurring in any offset* $\mathbf{n} \in N$ *as a component and* $r_i = i \bmod m$.
- *The numbering of neighbors is basically the same as for the* $\mathcal{A}_i$*:*

$$\nu_i(j) = \begin{cases} \nu_A(j) & \text{iff } j < n \\ \mathbf{r_i} & \text{iff } j = n \end{cases}$$

- *We define* $f_B : Q_B^{n+1} \to Q_B$ *by specifying the two resulting components separately:*

$$s(f_B(q_0, \ldots, q_n)) = f_i(s(q_0), \ldots, s(q_{n-1})) \text{ where } i = s(q_n) - s(q_0) \bmod m$$
$$p(f_B(q_0, \ldots, q_n)) = p(q_0)$$

Lemma 3. *$\mathcal{B}_i$ simulates $\mathcal{A}_i$ (for all $0 \le i < m$) in the following sense: For any $\mathcal{A}$-configuration c and all $t \ge 0$ one has*

$$E(\mathcal{A}_i^t(c)) = \mathcal{B}_i^t(E(c))$$

Proof. Consider an arbitrary i and an arbitrary $\mathcal{A}$-configuration c. The case $t = 0$ is trivial. Hence it suffices to prove the claim for $t = 1$, the rest follows by an easy induction. From the definition of $p(f_B(\ldots))$ immediately follows that the second component of a $\mathcal{B}$-cell does never change its value (since we always assume $\nu(0) = 0$). It remains to have a look at the first components. Consider an arbitrary cell $\mathbf{j} \in R$. By definition of E one gets

$$s(E(\mathcal{A}_i(c))(\mathbf{j})) = \mathcal{A}_i(c)(\mathbf{j}) = f_i(c(\mathbf{j} + \nu_A(0)), \ldots, c(\mathbf{j} + \nu_A(n-1)))$$

On the other hand for $c' = E(c)$ holds

$$\begin{aligned} s(\mathcal{B}_i(c')(\mathbf{j})) &= s(f_B(c'(\mathbf{j} + \nu_i(0)), \ldots, c'(\mathbf{j} + \nu_i(n-1)), c'(\mathbf{j} + \nu_i(n)))) \\ &= f_x(s(c'(\mathbf{j} + \nu_i(0))), \ldots, s(c'(\mathbf{j} + \nu_i(n-1)))) \\ &= f_x(s(c'(\mathbf{j} + \nu_A(0))), \ldots, s(c'(\mathbf{j} + \nu_A(n-1)))) \\ &= f_x(c(\mathbf{j} + \nu_A(0)), \ldots, c(\mathbf{j} + \nu_A(n-1))) \end{aligned}$$

By Lemma 1 the value x is the same for all cells $\mathbf{j}$ and by the definition of $s(f_B(q_0, \ldots, q_n))$ it is $x = s(q_n) - s(q_0) \bmod m = r_i - 0 \bmod m = i$.

Hence one gets exactly the same value as on the left hand side of the claim.

4 Simulating Infinitely Many CA Using One Local Function

In order to make it easier to describe the construction, only one-dimensional CA will be considered from now on. The generalization to the higher dimensional case is at most tedious but not difficult.

In this section we will describe CA $\mathcal{U}_i$ which can simulate any CA $\mathcal{A}_i$ having 2 states per cell, a neighborhood of arbitrary size and shape and an arbitrary

local function. As in Section 3, all $\mathcal{U}_i$ will only differ in the shape of their neighborhoods. The main difference to the previous construction is, that, in order to achieve universality, we now have to deal with *infinitely many* $\mathcal{A}_i$.

Also, the embedding of $\mathcal{A}$-configurations into $\mathcal{U}$-configurations will be different from the one in Section 3. But we will maintain the feature that the embedding is independent of the CA $\mathcal{A}_i$ to be simulated. We choose $Q_U = \{0,1\} \times Q'$ for some set Q' which contains the symbols $\circ$ and $\llcorner\!\lrcorner$ (among others). The following embedding $E : Q_A^R \to Q_U^R$ of configurations will be used:

$$E(c)(j) = \begin{cases} (c(j), \circ) & \text{if } j = 0 \\ (c(j), \llcorner\!\lrcorner) & \text{otherwise} \end{cases}$$

I.e., nothing is changed, except that a special marker $\circ$ is set in one cell. It doesn't matter which cell; we have chosen $j = 0$.

Conceptually, the work of any $\mathcal{U}_i$ consists of three phases:

1. A binary string representing the CA $\mathcal{A}_i$ to be simulated is generated.
2. The input representing the initial configuration for $\mathcal{A}_i$, is transformed.
3. $\mathcal{A}_i$ is simulated by $\mathcal{U}_i$.

As will be seen later on, the second and third phase are overlapped. In the following subsections we will sketch the most important aspects of the construction:

1. how the number of the CA to be simulated is generated as a binary string;
2. how the CA to be simulated is represented in this string;
3. how the initial configuration is transformed for simulation;
4. how one step of one cell can be simulated.

4.1 Computing the Representation of the Simulated CA

Consider the following CA: The set of states[1] is $Q_1 = \{0,1,\llcorner\!\lrcorner\} \times \{0,1,\llcorner\!\lrcorner\} \times \{\bullet, \circ, >, <, \llcorner\!\lrcorner\}$. We say that the cells consist of 3 *registers* containing "*sub-states*". Registers containing sub-state $\llcorner\!\lrcorner$ will be left empty and called *empty*. The first registers (shown at the top of each cell in Figure 1) are used for "sum bits", the second (in the middle) are used for "carry bits" and the third (at the bottom) are used for signals.

The neighborhood is $N = \{-r, -1, 0, 1, r\}$ where $r \geq 2$. The neighbor at position r is called the *"remote" neighbor* of the origin. It can identify itself, because it sees the marker $\circ$ at its neighbor at position $-r$.

We will explain the local rule with the help of Figure 1. The CA will be started in a configuration where all parts of all cells are empty except one cell. Without loss of generality assume that this is cell 0 and call it the *origin* cell from now on. Initially its first two registers are empty and the third contains $\circ$.

The goal is to reach a configuration which contains the number r, i.e. the distance of the remote neighbor, in binary representation in the first registers of some cells. To achieve this the CA does the following in parallel:

[1] Basically, this set Q_1 is part of the set Q' mentioned in the beginning of this section.

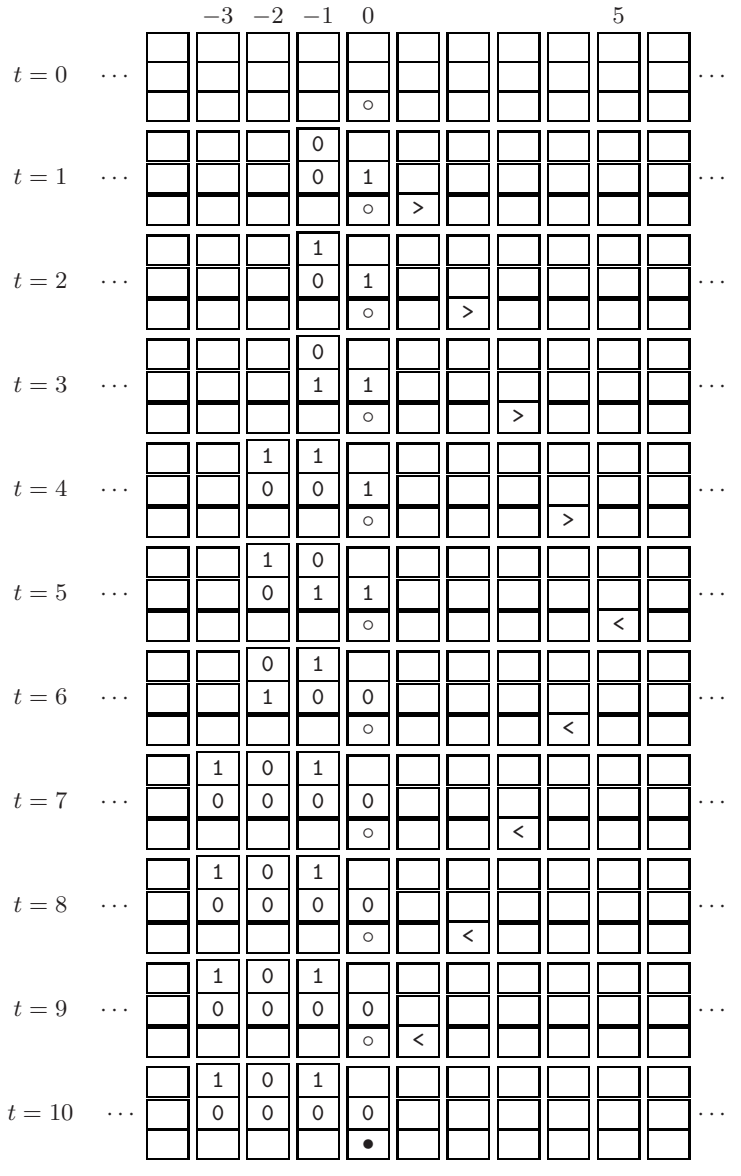


Fig. 1. A CA computing the distance of the remote neighbor in binary ($r = 5$)

- Using the third registers a signal (depicted > in Fig. 1) is started from the origin to the right. It is passed on to the next neighbor until it reaches the cell *which observes the state of the origin cell as its neighbor at* $-r$. There the signal reverses its direction (depicted <) and moves back to the origin where it changes ∘ to • after $2r$ steps indicating that everything has been done.

- Simultaneously to the left of the origin a counter is established, initialized with 0 and incremented in each step until the origin observes that the signal just described has arrived at its remote neighbor. Thus the final counter value is r. The time until the signal comes back to the origin is used to process the 1 carry bits in the counter cells if there are any.

4.2 Representation of a CA to Be Simulated

It is clear that each two-state CA can be represented as a word over some fixed alphabet. Without going into details let us simply give an example: As the alphabet one could use $\{\texttt{0}, \texttt{1}, \texttt{+}, \texttt{-}, \texttt{±}, \texttt{[}, \texttt{]}\}$.

As the complete encoding of a CA one can use the concatenation of the encodings of the neighborhood and the encoding of the local transition functions. For example, the neighborhood $\nu = (-2, 0, 1)$ can be encoded as

$$\texttt{[[-10][±0][+1]]}$$

For example, the local transition function of rule 90 can be encoded as

$$\texttt{[[0000][0011][0101][0111][1000][1011][1101][1110]]}$$

We call a sub-word of the form $\texttt{[}q_0, \ldots, q_{n-1}, f(q_0, \ldots, q_{n-1})\texttt{]}$ a *local rule*. If the i-th neighbor of a cell is in state s, we say that $\texttt{[}q_0, \ldots, q_{n-1}, f(q_0, \ldots, q_{n-1})\texttt{]}$ is *relevant*, iff $q_i = s$. For example, if the state of the second neighbor is `0`, the first, second, fifth and sixth local rule are relevant.

The CA $\mathcal{U}_i$ produce binary strings as representations. For that the symbols above have to be encoded as bit strings of some fixed length. It is clear that a CA can check whether an arbitrary binary string really is the encoding of a CA as described above quite easily.

4.3 Transforming the Configuration to Be Used for the Simulation

For the simulation each $\mathcal{U}$-configuration consists of *blocks*. Each block

- contains a complete copy of the description of $\mathcal{A}_i$ as explained in the previous subsection,
- is responsible for the simulation of one $\mathcal{A}_i$-cell,
- stores the current and the previous state of the represented $\mathcal{A}_i$-cell, and
- together with each state one of the integers $\{0, 1, 2\}$. They are used for Nakamura's technique [3] of simulating synchronous updating in a framework where the updates cannot really be synchronous.

After the first phase, $\mathcal{U}_i$ has one copy of the description of $\mathcal{A}_i$. The cells containing this copy form the block for simulating cell -1 of $\mathcal{A}_i$; we call this the "block -1". In the sequel $\mathcal{U}_i$ has to

- establish more and more subsequent blocks to the left and to the right with copies of the description of $\mathcal{A}_i$,

- shift the states of the $\mathcal{A}_i$ cells $-2, -3, -4, \ldots$ to the left and the states of the $\mathcal{A}_i$-cells $0, 1, 2, \ldots$ to the right to their corresponding blocks, and
- initialize the mod3 counters.

Since there is an infinite number of cells to be simulated, this process will never come to an end. But after a finite number of steps enough blocks will be set up so that the first step of cell -1 can be simulated. At some later time, there will be enough blocks to simulate the first step of each neighbor of cell -1. Hence afterwards the second step of cell -1 can be simulated, etc.

We assume that the reader is familiar enough with standard CA techniques, so that she/he could fill out the details here.

4.4 The Simulation

Whenever a block wants to simulate one state transition of the represented $\mathcal{A}_i$-cell, it does the following:

1. It sends some kind of state request signals to the "neighbor blocks", i.e., the blocks representing the neighbors of the cell to be simulated.
2. If such a block has already made enough simulations steps (according to Nakamura's technique), it can attach the needed state to the signal and send it back to its originating block. Otherwise the information is sent back, that the state is not yet available.
3. Upon arrival of a state in the block that had requested it, the corresponding rules of the transition table are marked as relevant. If a "not-yet-available" value returns, the request signal is sent again.
4. If all signals have returned a valid state, the new state can be read off the transition table and stored as the new current state while moving the old the previous register. The mod 3 counter of the new current state gets 1 plus the value of the now previous state.

The two basic technical aspects which deserve further explanation are the signals and how they are used to select the correct "row of the transition table".

The signals have to know how many blocks the have to travel (and they have to travel as many blocks back to their origin). One can use a standard signal of constant speed (smaller than 1 for the algorithms described below to work) and attach to it a pair of binary numbers (d, D), which initially are both the number of blocks the signal has to travel. While d is repeatedly used for counting down to 0, D is not changed but used to restore the original value for d.

The binary values can be arbitrarily large and hence have to be stored in a distributed fashion in some subsequent cells. When travelling to the neighbor block, d is decremented at each left (or right, depending on the direction of travel) block boundary it passes. The signal has reached its destination block when the value is 0. For the travel back d is reset to the initial value from D.

The selection of the relevant local rule, it is most convenient, to image that the signals just described have another pair of numbers (k, K) attached to them (in addition to d and D). It is the index of the neighbor: If initially $d = D = \nu(i)$,

then $k = K = i$. Analogously to the above, K is used to restore the original value of k, whenever k has been decremented to 0. This is done during the process of marking relevant local rules: Whenever a signal is returning from the K-th neighbor with some state s, it passes all local rules $[q_0, \ldots, q_{n-1}, f(q_0, \ldots, q_{n-1})]$. Each time it arrives at some q_i, it checks the content of k.

- If $k = 0$ and q_i is identical with s, q_i is marked as relevant.
- If $k > 0$, the value k is decremented and the signal moves on to q_{i+1}.

At the end] of the local rule, k is restored to K, so that the next local rule can be checked analogously. In addition, each such signal can always check, whether *all* states $q_0, \ldots, q_{n-1}$ are marked as relevant. If this is the case, the state at the end of the local rule is the new state of the simulated cell.

5 Summary and Outlook

In Section 3 we he have shown that it is possible to simulate a finite number of CA $\mathcal{A}_i$ with the same neighborhood and different local rules by CA $\mathcal{B}_i$ with the same local rule and different neighborhoods. In Section 4 we he have shown that it is even possible to simulate an infinite number of CA $\mathcal{A}_i$ with neighborhoods of different size and shape and different local rules by CA $\mathcal{U}_i$ with the same local rule and different neighborhoods which all have size 5.

In both constructions the embeddings of the configurations of the simulated CA into configurations of the simulating CA were independent of the specific CA to be simulated. It was *only* the difference in the neighborhoods of the simulating CA that could be and was exploited. The two constructions did make use of different types of embeddings of configurations and different types of simulations. It remains to be investigated, whether in the construction showing universality one really needs to "break symmetry" by setting the special marker in one cell.

It would not be the first such case. E.g., when looking at the simulation of arbitrary CA by reversible CA for infinite configurations, it is known that this is possible for some kind of simulation [1] while it is provably impossible for another type of simulation [2].

References

1. Durand-Lose, J.O.: Reversible space-time simulation of cellular automata. Theoretical Computer Science 246, 117–129 (2000)
2. Hertling, P.: Embedding cellular automata into reversible ones. In: Calude, C.S., et al. (eds.) Unconventional Models of Computation, pp. 243–256. Springer, Heidelberg (1998)
3. Nakamura, K.: Synchronous to asynchronous transformation of polyautomata. Journal of Computer and System Sciences 23, 22–37 (1981)

Hardware Modelling of Cellular Automata: The Game of Life Case

Juan A. Gómez-Pulido, Juan M. Matas-Santiago, Francisco Pérez-Rodríguez, Miguel A. Vega-Rodríguez, Juan M. Sánchez-Pérez, and Francisco Fernández de Vega

Department of Computer Sciences. University of Extremadura. Cáceres, Spain
jangomez@unex.es

Abstract. In this paper a study about the modelling, simulation and implementation on reconfigurable hardware of the Conway's Game of Life is shown. The Game of Life is a very popular case of cellular automata. The purpose of this study was to know the capability to model cellular automata by means of the high level hardware description language Handel-C, and to determine a first approximation to the hardware performance of the algorithm by means of the simulation of the implementation on reconfigurable FPGA devices. We can say, thanks to the found results, that the employed modelling technique allows us to prototype easily in short time this kind of algorithms, and the obtained simulation timing results are better than the found in a software version, up to a determined high number of iterations of the game.

Keywords: Cellular Automata. Reconfigurable Computing. FPGA.

1 Introduction

Reconfiguration of circuitry at runtime to suit the application at hand has created a promising paradigm of computing that blurs traditional frontiers between software and hardware. This powerful computing paradigm, named reconfigurable computing, is based on the use of programmable logic devices, mainly field programmable gate arrays (from now on, FPGAs) [1] incorporated in board-level systems. FPGAs have the benefits of the hardware speed and the software flexibility; also, they have a price/performance ratio much more favourable than ASICs (Application-Specific Integrated Circuits). For these reasons, FPGAs are a good alternative for many real applications in image and signal processing, multimedia, robotics, telecommunications, cryptography, networking and computation in general [2].

Furthermore, as the reconfigurable computing is becoming an increasingly important computing paradigm, more and more tools are appearing in order to facilitate the FPGA programmability using higher-level HDLs (Hardware Description Languages). In this line, several research projects have been developed to bring new high-level languages, most notably, SpecC [3], SystemC [4] or Ocapi-xl [5]. And on the other hand, several companies are proposing their own high-level HDLs, such as Handel-C [6] or Forge [7]. The main advantage of all these new hardware description languages is their simplicity, where the hardware design can be defined and evaluated using a

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 589 – 595, 2007.
© Springer-Verlag Berlin Heidelberg 2007

pseudo-C programming style. In this way, FPGAs are making it possible for thousands of computer engineers to have access to digital design technology in an easier way, obtaining a better performance with a similar flexibility to software [8].

In the simulations done in this work we use different FPGA devices, such us Xilinx Virtex 800 and VirtexII 2000.

2 The Purposed Work

Cellular Automata (from now on, CA) have been satisfactorily applied for solving optimisation problems. However, a CA used for solving a large problem may have a very high computational cost, even when it is run on a high-performance machine. For this reason new methods that increase the velocity of the algorithms are needed. We believe that implementing a CA directly on hardware may increase its efficiency (decreasing the running time) in such a way that we can tackle more complex problems. In this work we describe the hardware modelling, using high level hardware description languages to implement in FPGAs, of a CA for solving the Conway's Game of Life, because this simple case can show us the ability of the used tools to model kind of algorithms.

3 The Game of Life

The Game of Life (from now on, GL or simply 'Life') is a cellular automaton proposed by the British mathematician John Horton Conway in 1970 [9]. It is a very popular example of the Cellular Automata. In this section we summarize the main characteristics exposed more in depth in [10].

Ever since its publication, it has attracted much interest because of the surprising ways the patterns can evolve. Life is an example of emergence and self-organization. It is interesting to observe the way that complex patterns can emerge from the implementation of very simple rules. From a theoretical point of view, it is interesting because it has the power of a universal Turing machine: that is, anything that can be computed algorithmically can be computed within GL.

Life has a number of patterns which emerge from starting positions. Soon after publication several interesting patterns were discovered. Its popularity was helped by the fact that it came into being just in time for a new generation of inexpensive minicomputers which were being released into the market, meaning that the game could be run for hours on these machines which were otherwise unused at night.

3.1 Description

The "game" is actually a zero-player game [10], meaning that its evolution is determined by its initial state, needing no input from human players. It runs on a grid of squares ("cells") which stretches to infinity in all directions. Each cell has eight "neighbours", which are the cells adjacent to it, including diagonally. Each cell can

be in one of two states: it is either "alive" or "dead" (or "on" and "off"). The grid evolves in discrete time steps. The states of all of the cells at one time are taken into account to calculate the states of the cells one time step later. All of the cells are then updated simultaneously. The transitions depend only on the number of live neighbours.

3.2 Rules of Life

There are three criteria to be considered, in order to make the behaviour of the population unpredictable.

- There should be no initial pattern for which there is a simple proof that the population grows without limit.
- There should be initial patterns that apparently do grow without limit.
- There should be simple initial patterns that grow and change for a considerable period of time before coming to an end in several possible ways.
  The rules are the following ones:
- Any live cell with fewer than two neighbours dies of loneliness.
- Any live cell with more than three neighbours dies of crowd.
- Any dead cell with exactly three neighbours lives.
- Any live cell with two or three neighbours lives, unchanged, to the next generation.
- All births and deaths occur simultaneously. Together they constitute a single generation.

3.3 The Game

The basic idea [10] is to start with a simple configuration of living cells (organisms) placed on a 2D grid by various methods. This constitutes the first generation. Conway's "genetic laws" for births, deaths and survivals (the four rules above) are then applied to the pattern and the next generation pattern is placed accordingly. Generation by generation the "player(s)" observe the various patterns that emerge.

4 Modelling and Implementing Life

The followed process (Fig. 1) consists of two parts: hardware implementation and software implementation.

4.1 Hardware Implementation

The hardware modelling of the algorithm has been performed using Handel-C under the Celoxica DK 3.1 development environment. Handel-C is a high level hardware description language. It allows a hardware programming similar to the C programming language. We have programmed two codes:

- Debug code. The code works with 20x20 integer arrays. Each array is placed on the output channel (chanout), cell to cell, in all iterations of the algorithm, storing the array in the text file output.txt. Each cell may have value 0 (active or alive cell) or 1 (no-active or dead cell). Depending on the place of the cell and its neighbour cell values in a given time the cell will act according the life game rules. The algorithm execution on the debugging mode gives a data file. This file is processed later using a Borland C++ programmed interpreter for watching the cell in the screen under a Windows XP environment. The purpose of this interpreter is to validate visually the algorithm behaviour in such a mode that this validation will be a condition for programming of a code for the hardware implementation.
- Hardware implementation code. This code generates an EDIF format data file that is synthesized for a Xilinx Virtex 2000E-bg560-8 FPGA. This FPGA is placed on the RC1000 prototyping board. The synthesis process has been performed using Xilinx ISE 7.1 development environment. During the synthesis process, detailed information of operation times and frequencies has been obtained. This information is very useful for determining the exact execution time of the algorithm on the FPGA, for a given number of iterations.

4.2 Software Implementation

In order to determine the software implementation performance, a C code exactly carrying out the same operations made by the Handel-C code, has been programmed. This software version of the Game of Life algorithm has been compiled with GNU GCC compiler, and it has been executed on a personal computer with a Pentium IV 1.7 GHz processor.

The time analysis of the algorithm execution has been carried out using the Intel VTune tool. This utility gives the user running time, that is, the real time from the starting to the ending of the algorithm execution. In this measure the cycle spent by the operating system and by the input/output operations is included.

5 Results

The time analysis results of both implementations (hardware and software) have been compared with the purpose to establish the performance and the effectiveness of the hardware implementation. In Fig. 2 two cases for this analysis are shown. Each case corresponds to an implementation on a different FPGA.

We have tried to compare FPGAs and PCs of contemporary technologies. We have verified that the execution of the algorithm using the FPGAs is better up to a high number of iterations. From this number the PC obtains more performance. This is because the impact of operating system common tasks, that are constantly running, is reduced as much as more time spent the algorithm during its execution.

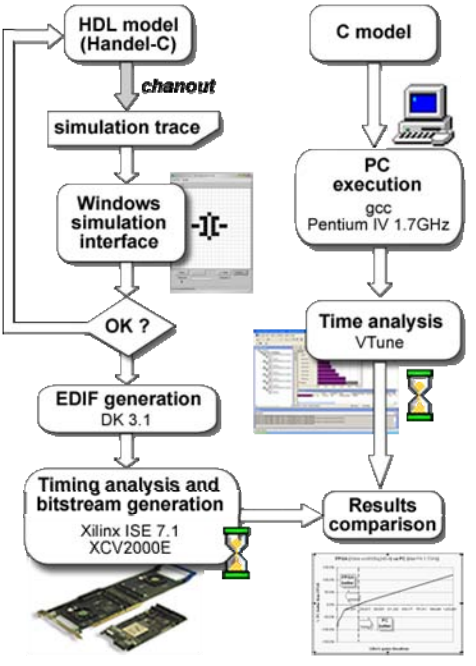


Fig. 1. Modeling, simulation and implementation followed process, and its comparison with the computer compiled algorithm. Two models have been programmed, one version for hardware implementation, and the other version to be executed in a general purpose processor.

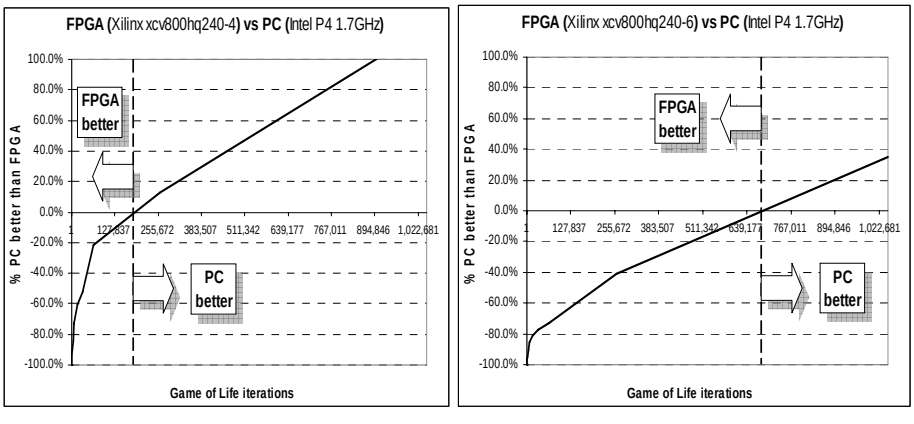


Fig. 2. Measures of how much is better the PC respect to the FPGA (%), for different number of iterations of the game

It is clear, as it can be seen comparing the two graphics of Fig. 2, as much better is the technology and performance of the used FPGA (device speed, power consumption,

Table 1. Efficiency results for different FPGAs

FPGA Xilinx Virtex	FPGA speed grade	Minimum period for clock	Maximum frequency	Iterations of Game of Life needed to reach the same timing cost than Pentium4 1.7 GHz
xcv800 -hq240	4	3.5 ns	285 MHz	180,000
xcv800 -hq240	6	2.7 ns	374 MHz	390,000
xcv2000e -bg560	8	1.8 ns	560 MHz	700,000

etc), greater is the iteration maximum number for which the FPGA is better than the PC. This fact is shown in Table I.

6 Conclusions

In this work we have tried to perform a first approximation to the hardware implementation effectiveness on FPGAs of a bio-inspired algorithm using Handel-C.

From the modelling point of view, we have seen the easy use of Handel-C for describing the algorithm. We observe that the developed code is minimally optimised for hardware implementation purposes, because it does not use concurrent sentences neither optimised arithmetic operations (using pipelining, as an example). So, we hope that an enhanced code carefully written will produce better results than those shown in this paper.

From the implementation point of view, we have verified the effectiveness of the algorithm running on the FPGA up to a high number of iterations; however, the PC is better when the algorithm exceeds that number.

These results encourage us to follow working on the code optimisation and in the use of the technologically more advanced FPGAs.

References

1. Zeidman, B.: Designing with FPGAs and CPLDs, CMP Books (2002)
2. Vega, M.A., Sánchez, J.M., Gómez, J.A.: Guest Editors' Introduction - Special Issue on FPGAs: Applications and Designs. Microprocessors and Microsystems 28(5-6), 193–196 (2004)
3. Gajski, D.D., Jianwen, Z., Dömer, R., Gerstlauer, A., Shuqing, Z.: SpecC: Specification Language and Methodology. Springer, Heidelberg (2000)
4. Arnout, G.: SystemC. In: Proceedings of the 2000 conference on Asia South Pacific design automation, pp. 573–578. ACM Press, New York (2000)
5. Vernalde, S., Schaumont, P., Bolsens, I.: An Object Oriented Programming Approach for Hardware Design. In: Proceedings of the IEEE Computer Society Workshop on VLSI 1999, p. 68. IEEE Computer Society, Los Alamitos (1999)
6. Ramamritham, K., Arya, K.: System Software for Embedded Applications. In: Proceedings of the 17th International Conference on VLSI Design, p. 22. IEEE Computer Society, Los Alamitos (2004)

7. (2005), http://www.xilinx.com
8. Vega, M.A., Sánchez, J.M., Gómez, J.A.: Advances in FPGA Tools and Techniques. In: Microprocessors and Microsystems, vol. 29(2-3), pp. 47–50. Elsevier Science, Amsterdam (2005)
9. Gardner, M.: Matematical games: The fantastic combinations of John Conway's new solitaire game life. Scientific American 223, 120–123 (1970)
10. (2006), http://www.wikipedia.com

Solving the Exploration's Problem with Several Creatures More Efficiently

Mathias Halbach and Rolf Hoffmann

TU Darmstadt, FB Informatik, FG Rechnerarchitektur
Hochschulstraße 10, D-64289 Darmstadt, Germany
Phone: +49 6151 16 {3713, 3606}, Fax: +49 6151 16 5410
{halbach, hoffmann}@informatik.tu-darmstadt.de

Abstract. We are presenting results of the creature's exploration problem with several creatures. The task of the creatures is to visit all empty cells in an environment with obstacles in shortest time and with a maximum of efficiency. The cells are arranged in a regular 2D grid and the underlying processing model is a Cellular Automaton (CA). We have investigated the question how many creatures and which algorithm should be used in order to fulfill the task most efficiently with lowest cost. We use a set of 10 different behaviors (algorithms) for the creature which have proved to be very efficient in the case where only one creature explores the environment. These algorithms were found by exhaustive search and evaluation by the aid of hardware (FPGA) implementation. Different environments and a varying number (1 to 64) of creatures were used in simulations in order to evaluate the cooperative work and efficiency. It turned out that for each environment a certain number of creatures and a certain algorithm is cost optimal in terms of work units. The total amount of work using one creature with the best algorithm X is many cases higher than the work using n creature with an adequate algorithm Y. Using several creatures, positive conflicts arise which may help to solve the problem more efficiently.

1 Introduction

We are presenting results from our project "Creature's exploration problem". The problem is the following: p creatures move around in an environment (cellular field) containing obstacles in order to visit all reachable empty cells in shortest time. All creatures behave according to the same rule.

The creature may perform four different actions:

- R (turn Right) with turn right only;
- L (turn Left) with turn left only;
- Rm (turn Right and move) with move forward and simultaneously turn right;
- Lm (turn Left and move) with move forward and simultaneously turn left.

A creature is only local dynamically connected to the neighbor in front of its moving direction. This cell is also called *front cell*.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 596–603, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The action *R/L* is performed if the front cell signals *not grant* because of an obstacle or a creature is in front, or because of a collision conflict. The action *Rm/Lm* is performed if the front cell signals *grant*. In case of a conflict in which two or more creatures want to move to the front cell, all creatures are blocked until the conflict disappears.

Conflicts are resolved in one phase (during the current clock cycle) which is different to other solutions where two phases are used. Each creature which wants to visit the same front cell sends a request signal and awaits a grant signal which is send back to the selected creature. In our current implementation no grant signal is send back if more than one creature requests. Our implementation is an extension of the CA model with von-Neumann neighborhood. Each cell contains logic (from the hardware view) or an additional function (from the mathematical view) which generates feedback signals for arbitration (fig. 1). By this technique the neighborhood of a creature (the front cell) is extended to all neighbors of the front cell. Therefore a creature can indirectly be informed that another creature is either two cells ahead or diagonal in front. The arbitration signal is only influencing the next state of the creature but is not further propagated to avoid very long propagation chains and possibly asynchronous oscillations.

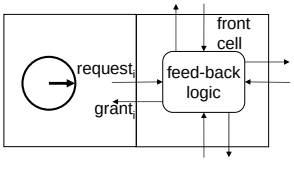


Fig. 1. Asynchronous feed-back logic used for arbitration

A creature is implemented with a *control machine* (MEALY automaton) and an *action machine* (MOORE automaton) which is controlled by the control machine (fig. 2a).

The behavior of the action machine is predefined and fixed. The state of the action machine is the direction r. The action machine reacts on the control signal d. If $d = 1$ the creature turns to the right ($r := r + 1$), otherwise to the left ($r := r - 1$).

The behavior of the control machine (also called algorithm for short in this context) is variable and can be configured by loading a state transition table. The control state is called s and the number of different states is n. Input to the state table is the control state s and the grant signal. Output of the state table is the control signal d and the next state s'. Note that the union of the control machine with the action machine results in a MOORE automaton.

An algorithm is defined by the contents of the table. We are coding an algorithm by concatenating the contents to a string line by line, e. g.
1L2L0L4R5R3R-3Lm1Rm5Lm0Rm4Lm2Rm // string representation
= 1L2L0L4R5R3R-3L1R5L0R4L2R // simplified string representation

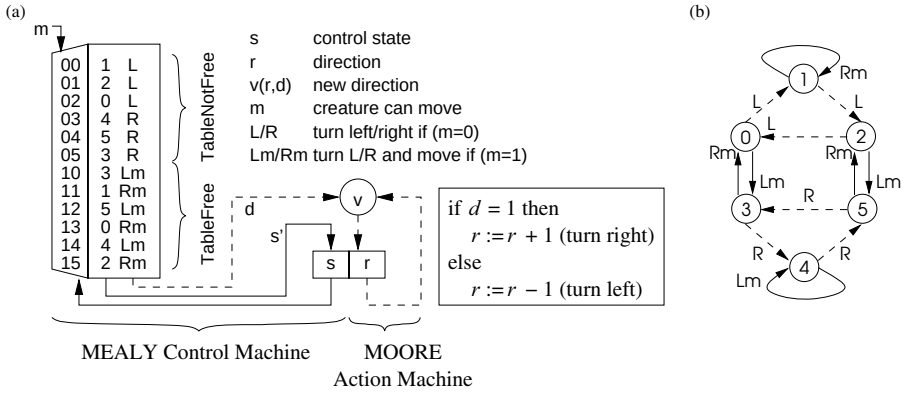


Fig. 2. (a) Table driven control machine and action machine; (b) corresponding 6-state algorithm G

The state table can be represented more clearly as a state graph (fig. 2b). If the state machine uses n states, we call such an algorithm *n-state algorithm*.

The number M of all algorithms which can be coded by a table oriented state machine is

$$M = (\#s \times \#y)^{(\#s \times \#x)}$$

where $n = \#s$ is the number of states, $\#x$ is the number of different input states and $\#y$ is the number of different output actions. Note that M increases dramatically, especially with $\#s$, which makes it very difficult or even impossible to check the quality of all algorithms in reasonable time for $\#s > 6$ with $\#x = \#y = 2$.

The following definitions and metrics are used

- $k :=$ number of creatures
- $R :=$ number of empty cells
- $g :=$ generation (time steps)
- $r(g) :=$ number of visited cells in generation g
- $r_{\max} :=$ the maximum number of cells which can be visited for $g \to \infty$
- $g_{\max} :=$ the first generation in which $r_{\max}$ is achieved
- $e := r_{\max}/R[\%]$, the *coverage* or *exploration rate*, i. e. $\frac{\text{visited cells}}{\text{all empty cells}}$,
- *successful* := true, if $e = 100\%$
- *speed* $:= R/g_{\max}$ (only defined for successful algorithms)
- *mean step rate* $:= \frac{1}{speed}$ (the mean number of cells visited in one generation)

In preceding investigation [13,14] we could discover the best 6-state algorithms for one creature. Using hardware support (FPGA technology) a large set of relevant algorithms was selected for 5 initial configurations by hardware enumeration, simulation and evaluation. From this set the best algorithms were selected during a software evaluation process applying 21 additional configurations.

The 10 best algorithms with respect to (1) success, (2) coverage and (3) speed are the following:

1. G: 1L2L0L4R5R3R-3L1R5L0R4L2R
2. B: 1R2R0R4L5L3L-3R1L5R0L4R2L
3. C: 1R2R0R4L5L3L-3R4R2L0L1L5R
4. A: 0R2R3R4L5L1L-1R5R4R0L2L3L
5. D: 1R2R3R1L5L1L-1R0L2L4R3L1L
6. E: 1R2L0R4L5L3L-3R4R5R0L1L2R
7. F: 1R2L0L4R5R3R-3L4L5L0R1L2R
8. H: 1L2L3R4L2R0L-2L4L0R3L5L4R
9. I: 1L2L3L4L2R0L-2L4L0R3R5L4R
10. J: 1R2R3R0R4L5L-4R5R3L2L0L1L

The goal of this investigation is to find out, how many creatures should be used in order to fulfill this task with a minimum of cost (work units).

The following additional measures were defined:

– The *cooperative work* of k creatures is proportional to the number of generations (time steps) to visit all cells, multiplied with the number of creatures.

$$W(k) = g_{\max}(k) \times k$$

– The *relative efficiency* is the work of one creature related to the work of k creatures using the same algorithm **Alg** for all the creatures.

$$F_{rel} = W_{Alg}(1)/W_{Alg}(k)$$

– The *absolute efficiency* is the work of one creature using the best algorithm $AlgBest$ divided by the work of k creatures using the algorithm **Alg**.

$$F_{abs} = W_{AlgBest}(1)/W_{Alg}(k)$$

2 Evaluation

We used the 10 best 6-state algorithms we had evaluated for one creature. Then we observed the global behavior using 1, 2, 4, 8, 12, 16, 28, 32, 60, 64 creatures in a field of size 33×33 (without the border cells). As an example 4 different environments (obstacle distributions) were chosen (fig. 3). All environments contain 129 obstacles and they are symmetric with respect to 90 degree rotation. Environment ENV2 uses a random distribution of the obstacles whereas the others have some particular structure.

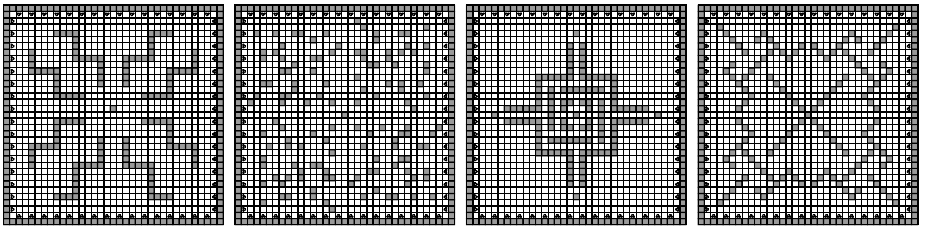


Fig. 3. Environments ENV1, ENV2, ENV3, ENV4

Table 1. Cooperative Work $W(k)$

ENV	k	A	B	E	F	H	J
1	1	9275		**7080**	**7217**		
1	2	10278	31254	20174			9098
1	4	14188	32952	10496	24744	96612	18376
1	8	8720	15168	17232	21608	71424	14256
1	12	13860	42060	16908	19308	28740	9684
1	16	**6720**	**10528**	22288	9200	**23904**	10608
1	28	13944	16296	14140	23296	31304	9548
1	32	9120	20896	17920	25472	28704	11136
1	60	20400	17880	18540	28140	43140	8820
1	64	10112	14336	27136	18048	32832	**8064**
2	1						
2	2						
2	4		**3244**	7252	**7224**		6084
2	8	12000	9240		8384		9256
2	12	**8676**	10428	8508	20232		10944
2	16	13904	9552	12784	13504	70304	14112
2	28	13552	11508	13272	7868	33208	**5124**
2	32	12576	6592	8320	10112	33216	8544
2	60	10380	8340	**6660**	9000	**20400**	17580
2	64	13824	7680	10816	18496	24128	9024
3	1	12797					
3	2	12086					
3	4	**10664**					
3	8	11776		29376	68504		20096
3	12	13080	114672		**13164**		9816
3	16	56784	52768	75120	65568		11168
3	28	25844	**29260**	171528	38024		**6384**
3	32	31456		134272	14784	249056	8416
3	60	31680	72660	52680	69360	**64320**	7020
3	64	81536	63872	**11520**	19584	114048	33408
4	1	18117					
4	2	15724					
4	4	**10828**					**8092**
4	8	27840			**8280**		15816
4	12	24972	80808	31284	13044	**6288**	17016
4	16	36320	32336	25520	28176	30272	37888
4	28	14952	13720	**12964**	26544	15316	18256
4	32	35264	**8736**	14240	8576	16832	15584
4	60	29580	16860	16380	22380	20460	11040
4	64	37696	12288	18752	41536	64064	29120

In the initial configuration the creatures were initially equally distributed along the borders.

The minimum number of work units (cost optimal) depends on the environment, the algorithm and the number of creatures (tab. 1). The cost optimal algorithm for ENV1 is algorithm A with 16 creatures (abbreviated A-16). The cost optimal algorithm is B-4 for ENV2, and is J-28 for ENV3, and is H-12 for ENV4.

The relative efficiency can only be computed if there exists an algorithm for one creature which is successful. All these cases are shown in tab. 2. The relative efficiency can be higher than 1.00 because several creatures can attack the problem from different start positions at the same time.

For the environment ENV3 the absolute efficiency can raise up to 2.00 for J-28 compared to A-1. This means that the cost to solve the problem with one creature

Table 2. Relative Efficiency F_{rel}

		Number of Creatures k									
Algorithm	ENV	1	2	4	8	12	16	28	32	60	64
A	1	1.00	0.90	0.65	**1.06**	0.67	**1.38**	0.67	**1.02**	0.45	0.92
E	1	1.00	0.35	0.67	0.41	0.42	0.32	0.50	0.40	0.38	0.26
F	1	1.00		0.29	0.33	0.37	0.78	0.31	0.28	0.26	0.40
A	3	1.00	**1.06**	**1.20**	**1.09**	0.98	0.23	0.50	0.41	0.40	0.16
A	4	1.00	**1.15**	**1.67**	0.65	0.73	0.50	**1.21**	0.51	0.61	0.48

Table 3. Absolute Efficiency F_{abs}

ENV	k	A	B	E	F	H	J
3	1	1.00					
3	2	**1.06**					
3	4	**1.20**					
3	8	**1.09**		0.44	0.19		0.64
3	12	0.98	0.11		0.97		**1.30**
3	16	0.23	0.24	0.17	0.20		**1.15**
3	28	0.50	0.44	0.07	0.34		**2.00**
3	32	0.41		0.10	0.87	0.05	**1.52**
3	60	0.40	0.18	0.24	0.18	0.20	**1.82**
3	64	0.16	0.20	**1.11**	0.65	0.11	0.38

Table 4. Amount of Conflicts per Creature

ENV	k	A	B	E	F	H	J
3	1	0.0	0.0	0.0	0.0	0.0	0.0
3	2	**0.0**	0.0	0.0	0.0	0.0	0.0
3	4	**0.0**	0.0	0.0	0.0	0.0	0.0
3	8	**0.5**	35.0	11.0	15.0	9.0	6.0
3	12	2.7	48.0	134.7	7.0	45.3	**2.3**
3	16	24.7	26.4	32.6	33.6	120.4	**4.6**
3	28	8.6	11.0	68.1	13.6	102.7	**2.6**
3	32	12.6	139.6	55.6	6.9	72.5	**3.5**
3	60	10.4	26.1	22.9	26.9	18.7	**2.5**
3	64	30.8	23.8	**3.4**	7.7	38.6	12.8

(A-1) is twice the cost to solve the problem with 28 creatures using algorithm J. This synergy effect is due to positive effects of conflicts and the distributed start positions. From tab. 4 can be seen that a certain amount of positive conflicts per creature, which is neither very low nor very high, contributes to a solution of lower cost.

3 Conclusion

The creature's exploration problem with several creatures was investigated with a respect to efficiency. The problem was modeled as a cellular automaton (CA) because CAs are massively parallel and they can be perfectly supported by hardware. The brain of the creature was modeled by a table driven control machine (holding the creature's algorithm) and a fixed action machine. We used the best 6-state algorithms which were discovered for the problem with one creature only. These algorithms were generated, simulated and evaluated by the

aid of special FPGA hardware because the number of the number of n-state algorithms is exploding with the number of states, inputs, and outputs. The ten best algorithms were used in the further evaluations using several creatures. It turned out that for a given environment and a certain algorithm a certain number of creatures is cost optimal with respect to work. Thus a super linear speedup can be reached using the right amount of creatures with an adequate algorithm. Moreover positive conflicts between creatures can be utilized to solve the whole task in shorter time.

References

1. Mesot, B., Sanchez, E., Pena, C.A., Perez-Uribe, A.: SOS++: Finding Smart Behaviors Using Learning and Evolution. In: Artificial Life VIII, p. 264. MIT Press, Cambridge (2002)
2. Koza, J.R.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
3. Halbach, M., Heenes, W., Hoffmann, R., Tisje, J.: Optimizing the Behavior of a Moving Creature in Software and in Hardware. In: Sloot, P.M.A., Chopard, B., Hoekstra, A.G. (eds.) ACRI 2004. LNCS, vol. 3305, pp. 841–850. Springer, Heidelberg (2004)
4. Halbach, M., Hoffmann, R.: Optimal Behavior of a Moving Creature in the Cellular Automata Model. In: Malyshkin, V. (ed.) PaCT 2005. LNCS, vol. 3606, pp. 129–140. Springer, Heidelberg (2005)
5. Halbach, M., Heenes, W., Hoffmann, R.: Implementation of the Massively Parallel Model GCA. In: Parallel Computing in Electrical Engineering (PARELEC), Parallel System Architectures (2004)
6. Halbach, M., Hoffmann, R.: Implementing Cellular Automata in FPGA Logic. In: International Parallel & Distributed Processing Symposium (IPDPS), Workshop on Massively Parallel Processing (WMPP), IEEE Computer Society, Los Alamitos (2004)
7. Hochberger, C.: CDL – Eine Sprache für die Zellularverarbeitung auf verschiedenen Zielplattformen. PhD thesis, TU Darmstadt (1998)
8. Hilbert, D.: Ueber die stetige Abbildung einer Linie auf ein Flachenstück. Mathematische Annalen 38(3), 459–460 (1891)
9. Peano, G.: Sur une courbe, qui remplit une aire plane. Mathematische Annalen 36(1), 157–160 (1890)
10. Halbach, M., Hoffmann, R.: Minimising the Hardware Resources for a Cellular Automaton with Moving Creatures. In: PARS Newsletter (2006)
11. Hoffmann, R., Ulmann, B., Völkmann, K.P., Waldschmidt, S.: A Stream Processor Architecture Based on the Configurable CEPRA-S. In: Grünbacher, H., Hartenstein, R.W. (eds.) FPL 2000. LNCS, vol. 1896, Springer, Heidelberg (2000)
12. Waldschmidt, S., Hochberger, C.: FPGA synthesis for cellular processing. In: IEEE/ACM International Workshop on Logic Synthesis, 9–55 – 9–63 (1995)
13. Halbach, M., Hoffmann, R., Both, L.: Optimal 6-State Algorithms for the Behavior of Several Moving Creatures. In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, Springer, Heidelberg (2006)
14. Hoffmann, R., Halbach, M.: Are several creatures more efficient than a single one? In: El Yacoubi, S., Chopard, B., Bandini, S. (eds.) ACRI 2006. LNCS, vol. 4173, pp. 707–711. Springer, Heidelberg (2006)

15. Fey, D., Schmidt, D.: Marching Pixels: A new organic computing principle for smart CMOS camera chips. In: Workshop on Self-Organization and Emergence - Organic Computing and its Neighboring Disciplines, Innsbruck, Austria, pp. 123–130 (2005)
16. Fey, D., Schmidt, D.: Marching-Pixels: A New Organic Computing Paradigm for Smart Sensor Processor Arrays. In: International Conference on Computing Frontiers 2005, pp. 1–7. ACM, Ischia, Italy (2005)

A New Time-Optimum Synchronization Algorithm for Two-Dimensional Cellular Arrays

Hiroshi Umeo and Hiroki Uchino

Univ. of Osaka Electro-Communication,
Neyagawa-shi, Hastu-cho, 18-8, Osaka, 572-8530, Japan
{umeo, uchino}@cyt.osakac.ac.jp

Abstract. The firing squad synchronization problem on cellular automata has been studied extensively for more than forty years, and a rich variety of synchronization algorithms have been proposed so far. In the present paper, we propose a new optimum-time algorithm for synchronizing two-dimensional cellular automata. The algorithm can synchronize any rectangular array of size $m \times n$ in optimum $m + n + max(m, n) - 3$ steps.

1 Introduction

We study a synchronization problem that gives a finite-state protocol for synchronizing a large scale of cellular automata. The synchronization in cellular automata has been known as a firing squad synchronization problem since its development, in which it was originally proposed by J. Myhill in Moore [1964] to synchronize all parts of self-reproducing cellular automata. The problem has been studied extensively for more than 40 years [1-14]. In the present paper, we propose a new optimum-time algorithm for synchronizing two-dimensional cellular automata. The algorithm proposed is designed based on a freezing-thawing technique developed by Umeo [2004]. It can synchronize any rectangular array of size $m \times n$ in optimum $m + n + max(m, n) - 3$ steps.

2 Firing Squad Synchronization Problem on Two-Dimensional Arrays

Figure 1 shows a finite two-dimensional (2-D) cellular array consisting of $m \times n$ cells. Each cell is an identical (except the border cells) finite-state automaton. The array operates in lock-step mode in such a way that the next state of each cell (except border cells) is determined by both its own present state and the present states of its north, south, east and west neighbors. All cells (*soldiers*), except the north-west corner cell (*general*), are initially in the quiescent state at time $t = 0$ with the property that the next state of a quiescent cell with quiescent neighbors is the quiescent state again. At time $t = 0$, the north-west corner cell C_{11} is in the *fire-when-ready* state, which is the initiation signal for

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 604–611, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the array. The firing squad synchronization problem is to determine a description (state set and next-state function) for cells that ensures all cells enter the *fire* state at exactly the same time and for the first time. The tricky part of the problem is that the same kind of soldier having a fixed number of states must be synchronized, regardless of the size $m \times n$ of the array. The set of states and next state function must be independent of m and n.

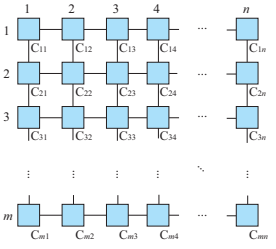


Fig. 1. A two-dimensional cellular automaton

The problem was first solved by J. McCarthy and M. Minsky who presented a $3n$-step algorithm for one-dimensional cellular array of length n. In 1962, the first optimum-time, i.e. $(2n-2)$-step, synchronization algorithm was presented by Goto [1962], with each cell having several thousands of states. Waksman [1966] presented a 16-state optimum-time synchronization algorithm. Afterward, Balzer [1967] and Gerken [1987] developed an eight-state algorithm and a seven-state synchronization algorithm, respectively, thus decreasing the number of states required for the synchronization. Mazoyer [1987] developed a six-state synchronization algorithm which, at present, is the algorithm having the fewest states for one-dimensional arrays. On the other hand, several synchronization algorithms on 2-D arrays have been proposed by Beyer [1969], Shinahr [1974], Szwerinski [1982] and Umeo, Maeda, Hisaoka and Teraoka [2006]. Beyer [1969] and Shinahr [1974] first proposed an optimum-time synchronization algorithm that can synchronize any 2-D array of size $m \times n$ in optimum $m+n+max(m,n)-3$ steps. Shinahr [1974] gave a 28-state implementation. Umeo, Hisaoka and Akiguchi [2005] presented a new 12-state synchronization algorithm operating in optimum-step, realizing a smallest solution to the rectangle synchronization problem at present.

3 Delayed Synchronization Scheme

First we introduce a *freezing-thawing* technique that yields a delayed synchronization algorithm for one-dimensional array.

Theorem 1. Let t_0, t_1, t_2 and Δt be any integer such that $t_0 \geq 0$, $t_1 = t_0+n-1$, $t_1 \leq t_2$ and $\Delta t = t_2 - t_1$. We assume that a usual synchronization operation is started at time $t = t_0$ by generating a special signal at the left end of one-dimensional array of length n. We also assume that the right end cell of the

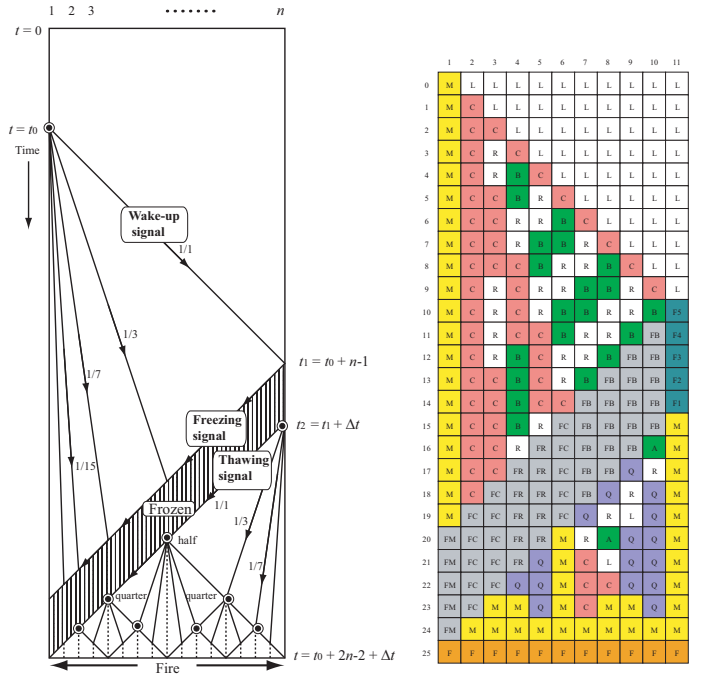


Fig. 2. Time-space diagram for delayed firing squad synchronization scheme based on the *freezing-thawing* technique (left) and delayed (for $\Delta t = 5$) configuration for optimum-time firing squad synchronization algorithm on $n = 11$ cells (right)

array receives another special signals from outside at time $t_1 = t_0 + n - 1$ and $t_2 = t_1 + \Delta t$, respectively. Then, there exists a one-dimensional cellular automaton that can synchronize the array of length n at time $t = t_0 + 2n - 2 + \Delta t$.

The array operates as follows:

1. Start an optimum-time firing squad synchronization algorithm at time $t = t_0$ at the left end of the array. A 1/1 speed, i.e., 1 cell per 1 step, signal is propagated towards the right direction to wake-up cells in quiescent state. We refer the signal as *wake-up signal*. A *freezing* signal is given from outside at time $t_1 = t_0 + n - 1$ at the right end of the array. The signal is propagated in the left direction at its maximum speed, that is, 1 cell per 1 step, and freezes the configuration progressively. Any cell that receives the freezing signal from its right neighbor has to stop its state-change and transmits the freezing signal to its left neighbor. The frozen cell keeps its state as long as no thawing signal will arrive.
2. A special signal supplied with outside at time $t_2 = t_1 + \Delta t$ is used as a *thawing* signal that thaws the frozen configuration progressively. The thawing signal

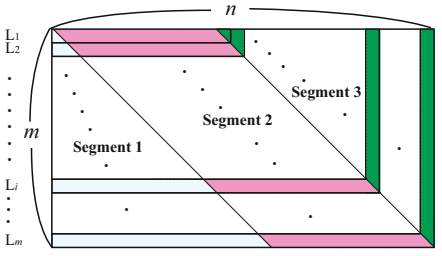


Fig. 3. A two-dimensional array of size $m \times n$ is regarded as consisting of m rotated (90° in counterclockwise direction) L-shaped 1-D arrays

forces the frozen cell to resume its state-change procedures immediately. See Fig. 2 (left). The signal is also transmitted toward the left end at speed 1/1.

The readers can see how those three special signals work. We can freeze the entire configuration during Δt steps and delay the synchronization on the array for Δt steps. Figure 2 (right) shows snapshots of delayed (for $\Delta t = 5$) configurations for optimum-time synchronization algorithm on eleven cells. In this example, note that the wake-up signal is supplied with the array at time $t = 0$. We refer the scheme as *freezing-thawing technique*. The technique was developed by Umeo [2004] for designing several fault-tolerant firing squad synchronization algorithms for one-dimensional arrays. In the next section, the freezing-thawing technique will be employed efficiently in the design of optimum-time synchronization algorithms for two-dimensional cellular arrays.

4 A New Optimum-Time Synchronization Algorithm for Two-Dimensional Cellular Arrays

In this section we develop a new optimum-time firing squad synchronization algorithm $\mathcal{A}$ operating in $m + n + \max(m, n) - 3$ steps for any array of size $m \times n$.

4.1 Segmentation and Initiation of Synchronization Process

Firstly, we assume that $m \leq n$. We regard a two-dimensional array of size $m \times n$ as consisting of m rotated (90° in counterclockwise direction) *L-shaped* one-dimensional arrays. Each L-shaped array is denoted by $\mathrm{L}_i, 1 \leq i \leq m$, shown in Fig. 3. Each L_i is divided into three segments, referred to as 1st, 2nd and 3rd segments of length i, $n-m$, and i, respectively. At time $t = 0$ a two-dimensional array M has a *General* at C_{11} and any other cells of the array are in quiescent state. The *General* generates three signals s_V, s_D and s_H, simultaneously, each propagating at 1/1-speed in the vertical, diagonal and horizontal directions, respectively. The s_V- and s_H-signals work for generating wake-up signals for the 1st and 3rd segments on each L-shaped array. The s_D-signal is used for printing

a delimiter of the 1st and 2nd segments. Their operations are as follows: The s_V-signal travels along the 1st column and reaches C_{m1} at time $t = m-1$. Then, it returns there and begins to travel at 1/2-speed along the 1st column towards C_{11}. On the return's way, the signal initiates the synchronization process for the 1st segment of each L_i. Thus a new *General* G_{i1} for the synchronization of the 1st segment of each L_i and its wake-up signal are generated at time $t = 3m - 2i - 1$ for $1 \leq i \leq m$. The s_D-signal travels along a diagonal line by repeating a zig-zag movement of going one cell to the right then going down one cell. Each time it visits a cell C_{ii} on the diagonal, it marks a special symbol to denote the delimiter of the 1st and 2nd segments of each L_i. The symbol on C_{ii} is marked at time $t = 2i - 2$ for any $i, 1 \leq i \leq m$. Note that the wake-up signal of the 1st segment of L_m knows its right end by the arrival of s_D-signal, where they meet at C_{mm} at the very time $t = 2m - 2$. The s_H-signal also travels along the 1st row at 1/1-speed and reaches C_{1n} at time $t = n - 1$. Then it reflects there and returns the same route at 1/2-speed. Each time it visits a cell of the 1st row on its return way, it generates a *General* G_{i3} at time $t = 2m + n - 2i - 1$ to initiate a synchronization for the 3rd segment on each $L_i, 1 \leq i \leq m$. The wake-up signal generated by G_{i1} reaches its right end at time $t = 2m - 2$ and generates a new *General* G_{i2} for the 2nd segment. The new general generates a wake-up signal. The wake-up signals for the 2nd and 3rd segments of L_i meet on $C_{i,n-m+i}$ at time $t = 2m + n - i - 2$. The collision of the two signals acts as a delimiter of the 2nd and 3rd segments in the case where $m \leq n$. Note that the synchronization operations on the 1st and 2nd segments are started at the left end of each segment. On the other hand, the synchronization on the 3rd segment is started at the right (upper) end of the segment.

4.2 Synchronization of L_m

Now we consider the synchronization on L_m. Figure 4 (left) shows a time-space diagram for synchronizing L_m. As was mentioned in the previous subsection, the synchronization of the 1st, 2nd and 3rd segments of L_m are started at time $t = m-1$, $t = 2m-2$ and $t = n-1$, respectively. Each General always generates a wake-up and a pre-thawing signal, each propagating at 1/1- and 1/2-speed in the same direction. The wake-up signal wakes up cells in the segment itself, however, the pre-thawing signal generates a thawing signal at its *neighbouring* segment that it encounters first. Precisely, the pre-thawing signal generated by G_{m1} reaches the left end of the 2nd segments at time $t = 3m - 3$. In the case $m \leq n$, the configuration of the 2nd segment have not been frozen yet, and the signal doesn't work. The pre-thawing signal generated by G_{m2} arrives at the delimiter between the 2nd and 3rd segments at time $t = 2n - 2$ and generates a thawing signal for the 3rd segment. In a similar way, the pre-thawing signal generated by G_{m3} initiates its thawing operation for the 2nd segment at time $t = 2m + n - 2$.

At time $t = 2m - 2$ the wake-up signal of the 1st segment reaches its right end and generates a freezing signal for the segment. Simultaneously, it initiates synchronization on the 2nd segments. The freezing signal for the 1st segment prop-

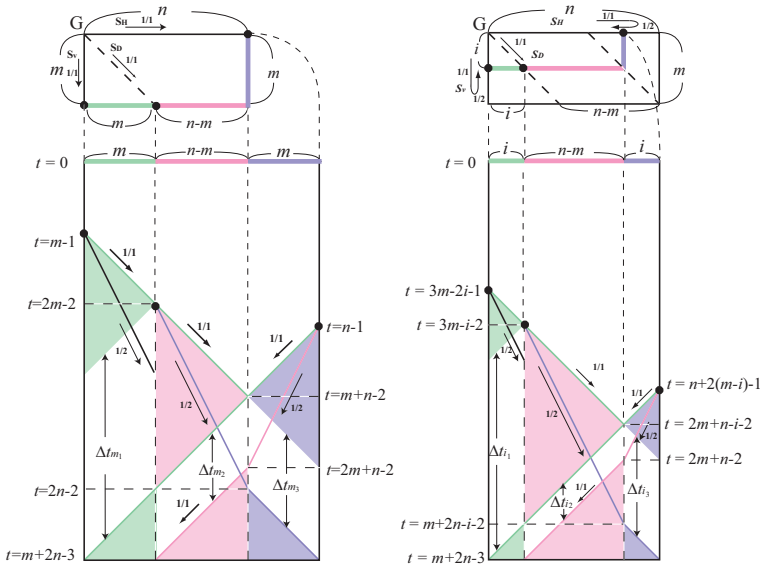


Fig. 4. Time-space diagram for synchronizing L_m (left) and L_i (right)

agates in left direction at 1/1-speed, and freezes the configuration on it. At time $t = 2n - 2$ a thawing signal is generated at its right end, that is initiated by the arrival of the freezing signal for the 2nd segment, which is explained again below. The wake-up signal for the 2nd and 3rd segments meets C_{mn} at time $t = m+n-2$, where C_{mn} acts as an end of the both two segments. A freezing signal is generated there for the 2nd and 3rd segments. It propagates in right and left directions at 1/1-speed to freeze the synchronization operations on the 2nd and 3rd segments, respectively. The freezing signal for the 2nd segment reaches at its left end at time $t = 2n - 3$, which generates a thawing signal for the 1st segment at time $t = 2n - 2$. The thawing signals for the 1st, 2nd and 3rd segments are generated at time $t = 2n - 2$, $t = 2m + n - 2$ and $t = 2n - 2$, respectively. Synchronization operations on each segment are delayed for $\Delta t_{m_j}, 1 \leq j \leq 3$ such that:

$$\Delta t_{m_j} = \begin{cases} 2(n-m) & j = 1 \\ m & j = 2 \\ n-m & j = 3 \end{cases} \tag{1}$$

The synchronization for the 1st segment is started at time $t = m - 1$ and its operations are delayed for $\Delta t = \Delta t_{m_1} = 2(n - m)$ steps. Now letting $t_0 = m - 1, \Delta t = \Delta t_{m_1} = 2(n - m)$ in [Theorem 1], the 1st segment of L_m can be synchronized at time $t = t_0 + 2m - 2 + \Delta t = m + 2n - 3 = m + n + max(m, n) - 3$. In a similar way, the 2nd and the 3rd segments can be synchronized at time $t = m + 2n - 3$. Thus, L_m can be synchronized at time $t = m + 2n - 3 = m + n + max(m, n) - 3$.

4.3 Synchronization of L_i

Now we discuss the synchronization for $L_i, 1 \leq i \leq m$. Figure 4 (right) shows a time-space diagram for synchronizing L_i. The wake-up signals for the three segments of L_i are generated at time $t = m + 2(m - i) - 1, 3m - i - 2$, and $n + 2(m - i) - 1$, respectively. Generation of freezing and thawing signals are done in a similar way as employed in L_m. Synchronization operations on each segments are delayed for $\Delta t_{i_j}, 1 \leq j \leq 3$ such that:

$$\Delta t_{i_j} = \begin{cases} 2(n-m) & j = 1 \\ i & j = 2 \\ n - m & j = 3 \end{cases} \tag{2}$$

The synchronization for the 1st segment of L_i is started at time $t = m+2(m-i)-1$ and its operations are delayed for $\Delta t = \Delta t_{i_1} = 2(n-m)$ steps. Now letting $t_0 = m + 2(m - i) - 1, \Delta t = \Delta t_{i_1} = 2(n - m)$ in [Theorem 1], the 1st segment of L_i can be synchronized at time $t = t_0 + 2i - 2 + \Delta t = m + 2n - 3$. In a similar way, the 2nd and the 3rd segments can be synchronized at time $t = m + 2n - 3$. Thus, L_i can be synchronized at time $t = m + 2n - 3 = m + n + max(m, n) - 3$.

In the case where $m > n$, a two-dimensional array of size $m \times n$ is regarded as consisting of n rotated L-shaped arrays. Segmentation and synchronization operations on each L-shaped array can be done almost in a similar way. It is noted that the right end cell of the 2nd segment works as a *General* for the 2nd segment. Any rectangle of size $m \times n$ can be synchronized at time $t = 2m + n - 3 = m + n + max(m, n) - 3$. Now we can establish the next theorem.

Theorem 2. The algorithm $\mathcal{A}$ can synchronize any $m \times n$ rectangular array in optimum $m + n + max(m, n) - 3$ steps.

5 Conclusions

We have proposed a new optimum-time synchronization algorithm for two-dimensional cellular arrays. The algorithm can synchronize any two-dimensional array of size $m \times n$ in optimum $m + n + max(m, n) - 3$ steps.

References

1. Balzer, R.: An 8-state minimal time solution to the firing squad synchronization problem. Information and Control 10, 22–42 (1967)
2. Beyer, W.T.: Recognition of topological invariants by iterative arrays. Ph.D. Thesis, p. 144. MIT, Cambridge (1969)
3. Hans-D, G.: Über Synchronisations - Probleme bei Zellularautomaten. Diplomarbeit, Institut für Theoretische Informatik, Technische Universität Braunschweig, p. 50 (1987)
4. Goto, E.: A minimal time solution of the firing squad problem. Dittoed course notes for Applied Mathematics 298, Harvard University, pp. 52–59 (1962)

5. Mazoyer, J.: A six-state minimal time solution to the firing squad synchronization problem. Theoretical Computer Science 50, 183–238 (1987)
6. Moore, E.F.: The firing squad synchronization problem. In: Moore, E.F. (ed.) Sequential Machines, Selected Papers, pp. 213–214. Addison-Wesley, Reading, MA (1964)
7. Shinahr, I.: Two- and three-dimensional firing squad synchronization problems. Information and Control 24, 163–180 (1974)
8. Szwerinski, H.: Time-optimum solution of the firing-squad-synchronization-problem for n-dimensional rectangles with the general at an arbitrary position. Theoretical Computer Science 19, 305–320 (1982)
9. Umeo, H.: A simple design of time-efficient firing squad synchronization algorithms with fault-tolerance. IEICE Trans. on Information and Systems E87-D(3), 733–739 (2004)
10. Umeo, H., Hisaoka, M., Akiguchi, S.: Twelve-state optimum-time synchronization algorithm for two-dimensional rectangular cellular arrays. In: Calude, C.S., Dinneen, M.J., Păun, G., Pérez-Jiménez, M.J., Rozenberg, G. (eds.) UC 2005. LNCS, vol. 3699, pp. 214–223. Springer, Heidelberg (2005)
11. Umeo, H., Hisaoka, M., Sogabe, T.: A survey on optimum-time firing squad synchronization algorithms for one-dimensional cellular automata. Intern. J. of Unconventional Computing 1, 403–426 (2005)
12. Umeo, H., Hisaoka, M., Teraoka, M., Maeda, M.: Several new generalized linear- and optimum-time synchronization algorithms for two-dimensional rectangular arrays. In: Margenstern, M. (ed.) MCU 2004. LNCS, vol. 3354, pp. 223–232. Springer, Heidelberg (2005)
13. Umeo, H., Maeda, M., Hisaoka, M., Teraoka, M.: A state-efficient mapping scheme for designing two-dimensional firing squad synchronization algorithms. Fundamenta Informaticae 74, 603–623 (2006)
14. Waksman, A.: An optimum solution to the firing squad synchronization problem. Information and Control 9, 66–78 (1966)

3D Motion Estimation Using a Combination of Correlation and Variational Methods for PIV

L. Alvarez, C.A. Castaño, M. García, K. Krissian, L. Mazorra, A. Salgado, and J. Sánchez

Departamento de Informática y Sistemas, Universidad de Las Palmas de Gran Canaria, Campus de Tafira s/n, 35017 Las Palmas de Gran Canaria, Spain
{lalvarez,ccastano,mgarcia,krissian,lmazorra, asalgado,jsanchez}@dis.ulpgc.es

Abstract. Estimation of motion has many applications in fluid analysis. Lots of work has been carried out using Particle Image Velocimetry to design experiments which capture and measure the flow motion using 2D images. Recent technological advances allow capturing 3D PIV image sequences of moving particles. In this context, we propose a 3D motion estimation technique based on the combination of an iterative cross-correlation technique and a variational (energy-based) technique. The performance of the proposed technique is measured and illustrated using numerical simulations.

1 Introduction

"Particle Image Velocimetry (PIV) is a technique which allows one to record images of large parts of flow fields in a variety of applications in gaseous and liquid media and to extract the velocity information out of these images" [6]. Once the flow motion has been captured, software tools are needed to evaluate and display the flow motion.

In this paper, we propose a technique for 3D fluid motion estimation applied to 3D-PIV. The most widely used technique for motion estimation in 2D-PIV is based on local correlation between two rectangular regions of the two images (see for instance [7]). This technique has a straightforward extension to 3D images. Another approach to motion estimation widely used in optical flow is a variational approach based on an energy minimization where on the one hand, we assume the conservation of the intensity of the displaced objects (in our case the particles) and on the other hand, we assume a certain regularity of the obtained flow. A variational approach was proposed in [4] in the context of 2D PIV. We propose to compare and combine both approaches in order to improve the accuracy of the flow estimation. The proposed method is very general and can be used in many applications of 3D flow estimation.

The paper is organized as follows: in section 2, we briefly describe the motion estimation using local cross-correlation; in section 3, we describe our variational approach; in section 4, we present the numerical experiments followed by the conclusion.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 612–620, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Motion Estimation Using Local Cross-Correlation

Cross-correlation is the most common technique for fluid motion estimation in PIV and is described, for instance, in [6]. We will denote I_1 and I_2 the two images from which we compute the motion $\mathbf{u}$, N the image dimension (in our case $N = 3$) and Ω the domain of definition of the images.

2.1 Basic Principle

Having the two volumes I_1 and I_2, for each voxel $\mathbf{v} = (v_x, v_y, v_z)$ of I_1, the method takes a rectangular subvolume $I_{1,\mathbf{v}}$ of I_1 centered on $\mathbf{v}$, and looks for a similar subvolume of I_2 centered on a neighbor $\mathbf{v} + \mathbf{d}$ of $\mathbf{v}$. The similarity measure between two rectangular subvolumes of the same dimensions is based on 2D cross-correlation and is defined as:

$$C_{\mathbf{v}}(I_1, I_2)(\mathbf{d}) = \sum_{\mathbf{y}=(-a,-b,-c)}^{(a,b,c)} I_1(\mathbf{v}+\mathbf{y})\ I_2(\mathbf{v}+\mathbf{d}+\mathbf{y}) \tag{1}$$

The voxel $\mathbf{v}$ is assigned the displacement $\mathbf{d}$ which gives the maximal value of the cross-correlation. Doing this for every voxel we obtain a complete vector field $\mathbf{u}$.

2.2 Implementation Using Fast Fourier Transform

Because the process of computing the cross-correlation for many subvolumes of I_2 and for each voxel is computationally heavy, the implementation takes advantage of the properties of the Fourier transform to improve the processing time. The Fourier transform has the property that a correlation in the spatial domain is equivalent to a multiplication in the Fourier domain.

$$C_{\mathbf{v}}(I_1, I_2) = \mathcal{F}^{-1}(\widehat{I_{1,\mathbf{v}}}\ \widehat{I_{2,\mathbf{v}}}^{*}), \tag{2}$$

where $I_{1,\mathbf{v}}$ is a rectangular subvolume of I_1 centered on the voxel $\mathbf{v}$, $\widehat{I_{1,\mathbf{v}}}$ is the Fourier Transform of the subvolume $I_{1,\mathbf{v}}$, the operator * denotes the complex conjugate, and $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. The image $C_{\mathbf{v}}(I_1, I_2)(\mathbf{d})$ gives the result of cross-correlation for all displacements $\mathbf{d}$ and the maximal value is a best estimate of the local displacement. Because of the hypothesis of periodicity introduced by the Fourier Transform, the window is usually chosen four times bigger than the expected displacement. The method is then extended to allow subvoxel accuracy by means of local interpolation of a Gaussian function close to the discrete maximum. When the correlation has been computed for every voxel, some kind of data validation procedure is needed to remove outliers.

The whole process should be applied iteratively a few times using the current result as an initialization for the next iteration. The iterative process can be initialized with a null vector field $\mathbf{u}^0 = 0$, and $\mathbf{u}^{n+1}$ can be estimated at each voxel of the lattice using the displacement with maximal correlation for a window of I_2 displaced by $\mathbf{u}^n$:

$$C_{\mathbf{v}}(I_1, I_2, \mathbf{u}^n) = \mathcal{F}^{-1}(\widehat{I_{1,\mathbf{v}}}\ \widehat{I_{2,\mathbf{v}+\mathbf{u}^n(\mathbf{v})}}^{*}), \tag{3}$$

By doing this, we can improve the accuracy of the fluid motion estimation. It also permits the progressive reduction of the size of the correlation window.

3 Variational Approach

Variational approach to motion estimation are often used for optical flow computation [5,3,2]. It consists in minimizing an energy as a function of the displacement and that depends on a pair of images I_1 and I_2.

In this section, E will denote the energy functional to minimize. For a given 3D vector field $\mathbf{u} = (u^x, u^y, u^z)^t$, the norm of its gradient $\| \nabla \mathbf{u} \|$ is defined as $\sqrt{\| \nabla u^x \|^2 + \| \nabla u^y \|^2 + \| \nabla u^z \|^2}$, and the Laplacian $\Delta \mathbf{u} = div(\nabla \mathbf{u})$ is defined as $(\Delta u^x, \Delta u^y, \Delta u^z)^t$.

The energy to minimize is expressed as :

$$E(\mathbf{u}) = \underbrace{\int_\Omega \left(I_1(\mathbf{x}) - I_2\left(\mathbf{x} + \mathbf{u}(\mathbf{x})\right)\right)^2 d\mathbf{x}}_{data\ term} + \underbrace{\alpha \int_\Omega \|\nabla \mathbf{u}(\mathbf{x})\|^2 d\mathbf{x}}_{regularization\ term}, \tag{4}$$

where α is a scalar coefficient that weights the smoothing term. Under the assumption of intensity conservation for each voxel, the first term (*data term*) becomes zero when the first image matches the second one displaced by $\mathbf{u}$: $I_1(\mathbf{x}) = I_2(\mathbf{x} + \mathbf{u}(\mathbf{x}))$. This term tries to find the vector field that best fits the solution. The second term is a *regularization term* which smoothes the vector field. There are a lot of ways to define the regularization term, including, for instance, discontinuities preserving constraints, etc. Since we deal with rather smooth flows, we have used the simplest regularity term presented above.

Euler-Lagrange equations yield:

$$(I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{u})).\nabla I_2(\mathbf{x} + \mathbf{u}) + \alpha div(\nabla \mathbf{u}) = 0 \tag{5}$$

3.1 Numerical Scheme

We propose to look for the minimum of the energy by solving (5) directly using a fixed point approach. An alternative is to use a gradient descent with either explicit or semi-implicit scheme. We use an iterative method to find $\mathbf{u}$:

$$\begin{cases} \mathbf{u}^0 & = \mathbf{u_0} \\ \mathbf{u}^{n+1} & = \mathbf{u}^n + \mathbf{h}^{n+1} \end{cases} \tag{6}$$

where we update the vector field $\mathbf{u}$ at each iteration by adding another vector field $\mathbf{h}$ with small displacements. The displacement $\mathbf{h}$ being small, we can use first order Taylor expansions of I_2 and ∇I_2 at $\mathbf{x} + \mathbf{u}^n$ to linearize (5):

$$d\mathbf{g} - \left[\mathbf{g}\mathbf{g}^t - dH'\right] \mathbf{h} + \alpha div(\nabla \mathbf{u}^n + \nabla \mathbf{h}) = 0 \tag{7}$$

denoting:

$$\mathbf{g}(\mathbf{x}) = \nabla I_2(\mathbf{x} + \mathbf{u}^n) \tag{8}$$

$$d(\mathbf{x}) = I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{u}^n) \tag{9}$$

$$H'(\mathbf{x}) = H(I_2)(\mathbf{x} + \mathbf{u}^n). \tag{10}$$

In the last equality, $H(I_2)(\mathbf{x})$ denotes the Hessian matrix of I_2 at the location $\mathbf{x}$. The term in second order spatial derivatives is usually neglected, supposing that the image varies slowly. Then, (7) becomes:

$$d\mathbf{g} + \alpha div(\nabla \mathbf{u}^n) - \mathbf{g}\mathbf{g}^t\mathbf{h} + \alpha div(\nabla \mathbf{h}) = 0 \tag{11}$$

After discretization using finite differences, the operator $div(\nabla \mathbf{h})$ can be divided in two terms $-2N\, I\, \mathbf{h}$ and $S(\mathbf{h})$, where the N is the image dimension and I is the identity matrix. The first term only depends on values of $\mathbf{h}$ at the current position $\mathbf{x}$ and the second term only depends on values of $\mathbf{h}$ at neighbor positions of $\mathbf{x}$: the vector $S(\mathbf{h})$ is written:

$$S(\mathbf{h}) = \begin{pmatrix} \sum_{\mathbf{y} \in N^*(\mathbf{x})} h^x(\mathbf{y}) \\ \sum_{\mathbf{y} \in N^*(\mathbf{x})} h^y(\mathbf{y}) \\ \sum_{\mathbf{y} \in N^*(\mathbf{x})} h^z(\mathbf{y}) \end{pmatrix}, \tag{12}$$

where $N^*(\mathbf{x})$ denotes the direct neighbors of $\mathbf{x}$ (4 in 2D and 6 in 3D), and $\mathbf{h} = (h^x, h^y, h^z)^t$.

Using $\mathbf{h}^{n+1}$ for the current location $\mathbf{x}$ and $\mathbf{h}^n$ for its neighbors, (11) becomes $A\mathbf{h}^{n+1} = b$, with $A = \mathbf{g}\mathbf{g}^t + \alpha 2N\, I$, and $b = d\mathbf{g} + \alpha div(\nabla \mathbf{u}^n) + S(\mathbf{h}^n)$. The matrix A is real, symmetric and positive definite, so it can be inverted and we can compute for each position $\mathbf{x}$, $\mathbf{h}^{n+1} = A^{-1}b$. To improve the convergence rate, we use a Gauss-Seidel method which updates the displacement $\mathbf{h}^{n+1}$ at position $\mathbf{x}$ using the values of $\mathbf{h}^{n+1}$ already calculated. This scheme is recursive and to avoid privileging the direction of scanning the image, we apply two successive iterations of Gauss-Seidel in reverse directions. Furthermore, we use a pyramidal approach to compute the displacement flow at several scales, using the results from a given scale to initialize to the following higher scale.

4 Experiments and Results

In this section, we present experiments on synthetic data using both methods (correlation and variational). We used a 3D flow based on realistic flow models to check the performance of the proposed methods. In these experiments, we first apply the correlation method to obtain a good approximation of the flow and then we refine the results with the variational approach.

4.1 Choice of the Parameters

The cross-correlation parameters are the window size in each dimension and the lattice spacing. The window size is approximately set to four times the expected

maximal displacement and is the same in each dimension. In the following experiments, we use a lattice spacing of 2 voxels in each dimension, and the final result is interpolated to obtain a dense estimation. The variational approach uses the parameters of α and the number of scales for the pyramidal approach. In the following experiments, we will show the sensitivity of the method to the regularization coefficient α and we will choose the value that gives best results.

4.2 Description of the Models

In the first model we use an incompressible 3D flow model suggested to us by Professor Scarano that can be found in [9] (section 3-9.2). It corresponds to the Stokes's solution for an immersed sphere. The flow moves in the horizontal axis direction with a velocity $(U, 0, 0)$, and it avoids the sphere located at the center of the volume. The flow inside the sphere is null.

The second model has been suggested to us by the CEMAGREF Institute and it has been obtained using a Large Eddy Simulation of the incompressible Navier-Stokes equations which defines the turbulent motion after a cylinder. It simulates a volume with synthetic particles following the horizontal axis and a cylinder situated on the z-axis obstructing the flow perpendicularly. We use two successive images from this sequence. The original model is a volume of 960 x 960 x 144 voxels but we limit our experiment to a window of 256 x 64 x 64 voxels to reduce the computation time. This window includes part of the cylinder and the turbulence behind it.

4.3 Experiments with Model 1 (Sphere)

Table 1 shows the best average error and standard deviation reached for each method individually and their combination. The correlation was applied 11 times with a window size of 8 voxels. The individual variational approach was applied using 3 scales and $\alpha = 8.4$. Finally, the combined method used the correlation result as an initialization for the variational method, applied with $\alpha = 1.1$ and one scale. The standard deviation is always about the same order as the average error, and although the variational and the cross-correlation methods reaches similar accuracies, combining them brings the error almost to half. Figure 1 (right) shows the final average error distribution using the combined scheme. We can observe that the highest error is located at the sphere boundaries.

The left curve in Figure 2 displays the average error evolution using the combined scheme. First, we apply 11 iterations of correlation technique. Next, we use the output flow provided by the correlation as the input flow of the variational technique (curve after iteration 11). We observe a significative improvement in the flow estimation error after using the variational method. The right curve depicts the average error for different values of α using the variational approach initialized with the best correlation result. The minimal average error (0.0138 voxel) is obtained for $\alpha = 1.1$.

Table 1. Comparison between the two methods for the different models

Model 1	**Correlation**	**Variational**	**Cor.+Var.**
av. error	0.0272	0.0245	0.0138
std. dev.	0.0275	0.0230	0.0169
Model 2	**Correlation**	**Variational**	**Cor.+Var.**
av. error	0.1673	0.0954	0.0694
std. dev.	0.1337	0.0849	0.0756

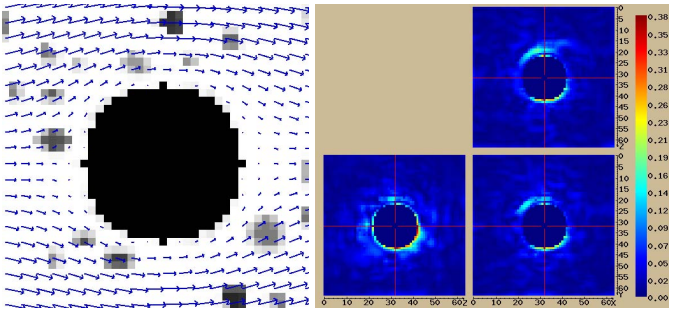


Fig. 1. Left, real flow (with zoom). Right, final error distribution (combined scheme).

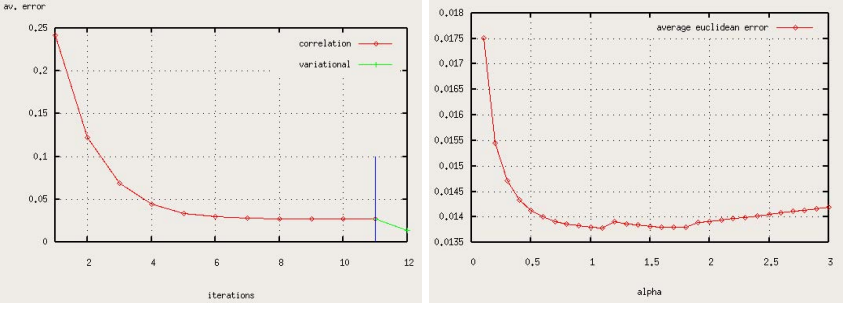


Fig. 2. Left, average error evolution using the combined scheme (11 times cor.+ var.). Right, average error depending on α for the combined approach.

4.4 Experiments with Model 2 (Cylinder)

We ran the same kind of experiments for this model as for the previous one. Table 1 shows the best average error and standard deviation reached for each method individually and their combination. The correlation was applied 6 times with a sequence of different window sizes: 16, 16, 8, 8, 4, 4. The variational approach was applied using $\alpha = 3.7$ and 3 scales. Finally, the combined method used the correlation result as an initialization for the variational method, applied with

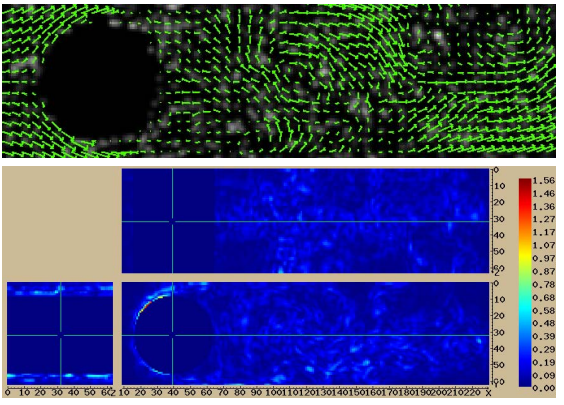


Fig. 3. Top, real flow. Bottom, final error distribution (combined scheme).

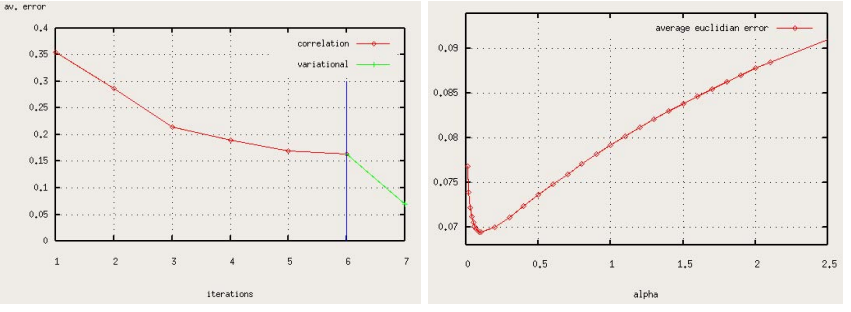


Fig. 4. Left, average error evolution using the combined scheme (6 times cor.+var.). Right, average error depending on α for the combined approach.

α=0.1 and one scale. The standard deviation is about the same order as the average error. In this experiment, the variational method reaches a better result than the correlation, and their combination still brings a substantial improvement. Figure 3 (right) shows the final average error distribution using the combined scheme. As in the previous model, the highest error is also located at the obstacle boundaries.

5 Conclusion

In this paper, we presented a 3D flow estimation technique based on the combination of correlation and variational methods. We have implemented both techniques and we have shown in the numerical experiments that a combination of both methods (using the output of the correlation technique as the initial

input of the variational method) improves the accuracy of the flow estimation. Indeed, several iterations of the cross-correlation technique followed by single scale variational approach gave us the best results in both images with a substantial improvement. This result can be interpreted in the following way: although the variational method is minimizing a global energy, it can only find a local minimum with a small displacement. Thus, the correlation technique gives the variational method a better initialization than the pyramidal approach.

Although we focused our attention to 3D fluid flow analysis, the proposed methodology is very general and can be applied to different application fields. Correlation based techniques and energy minimization techniques have been developed in the research community in a completely independent way. Each one has its own advantages and limitations but we think that an adequate combination of both can improve the global estimation of the flow.

In future work, we plan to investigate other regularization terms as proposed in [8,1]. We also think that including physical 3D flow constraints, as for instance the incompressibility, to the 3D flow estimation, is a very important issue that allows combining the mathematical models of fluid motion with the experimental data. We plan to compare our current method with approaches which include an incompressibility constraint within the variational formulation [4,10] and also apply solenoidal projections to make our results divergence free.

Acknowledgments

This work has been funded by the European Commission under the Specific Targeted Research Project FLUID (contract no. FP6-513663). We acknowledge the Research Institute CEMAGREF (France) for providing to us the PIV sequence we have used in the experiments.

References

1. Alvarez, L., Weickert, J., Sánchez, J.: Reliable estimation of dense optical flow fields with large displacements. International Journal of Computer Vision 39(1), 41–56 (2000)
2. Barron, J.L., Fleet, D.J., Beauchemin, S.S.: Performance of optical flow techniques. IJCV 12(1), 43–77 (1994)
3. Beauchemin, S.S., Barron, J.L.: The computation of optical flow. ACM Computing Surveys 27(3), 433–467 (1995)
4. Corpetti, T., Heitz, D., Arroyo, G., Mémin, E., Santa-Cruz, A.: Fluid experimental flow estimation based on an optical-flow scheme. Experiments in Fluids 40(1), 80–97 (2006)
5. Horn, B., Schunck, B.: Determining optical flow. MIT Aritificial Intelligence Laboratory (April 1980)
6. Raffel, M., Willert, C., Kompenhans, J.: Particle Image Velocimetry. A Practical Guide. Springer, Heidelberg (1998)
7. Scarano, F.: Iterative image deformation methods in piv. Measc. Sci. Technol. 13, R1–R19 (2002)

8. Weicker, J., Schnorr, C.: A theoretical framework for convex regularizers in pde-based computation of image motion. International Journal of Computer Vision 45(3), 245–264 (2001)
9. White, F.: Viscous Fluid Flow. McGraw-Hill, New York (2006)
10. Yuan, J., Ruhnau, P., Memin, E., Schnörr, C.: Discrete orthogonal decomposition and variational fluid flow estimation. In: Kimmel, R., Sochen, N.A., Weickert, J. (eds.) Scale-Space 2005. LNCS, vol. 3459, pp. 267–278. Springer, Heidelberg (2005)

Helicopter Flight Dynamics Using Soft Computing Models

Javier de Lope[1], Juan José San Martín[2], and José A. Martín H.[3]

[1] Dept. of Applied Intelligent Systems (UPM)
javier.delope@upm.es
[2] Microbótica, S.L.
peco@microbotica.es
[3] Dep. de Sistemas Informáticos y Computación (UCM)
jamartinh@fdi.ucm.es

Abstract. In this paper we propose a novel approach for the control of helicopters, using a flight dynamics model of the aircraft to develop reliable controllers by means of classical procedures, evolutionary either reinforcement learning techniques. Here we are presenting the method that we use to estimate the aircraft position, including the low level image processing, the hardware configuration which allows us to register the commands generated by an expert pilot using a conventional radio control (RC) transmitter, and how both variables are related by an artificial neural network (ANN).

1 Introduction

Nowadays the control of helicopter flight is one of the most challenging control problems. Helicopters have complex and noisy dynamics that makes difficult to create simple and suitable controllers. Helicopters maneuver in a very unstable medium which is altered by air currents sometimes generated by their own propellers. On the other hand, the helicopter's inertia makes hard to take the control of its movements. A great degree of anticipation is needed to make the maneuvers, including the apparently most simple ones such as hovering in place.

Probably the first successful attempt to control an unmanned helicopter was carried out by Sugeno [1]. Sugeno developed a fuzzy-logic-based control system that enabled simple commands to the helicopter such as *take off* or *turn left* to replace the aircraft's normal inputs. They used a complete array of sensors which includes gyros and accelerometers to measure three dimensional accelerations, velocities, attitude angles and angular velocities and a differential GPS to obtain the three dimensional position and velocities achieving a reasonable accuracy.

Recently other autonomous control systems have been developed. Corke *et al.* [2,3] developed a low-cost flight control system for small helicopters. Their approach differs from others in not using expensive inertial and GPS system. They use low-cost inertial sensors and a pair of CMOS cameras aboard the helicopter, in order to estimate the vehicle height and motion at 10 Hz from images of natural terrain with no artificial landmarks.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 621–628, 2007.
© Springer-Verlag Berlin Heidelberg 2007

A different approach but using also a vision system to estimate the aircraft position has been developed at Stanford University [4]. They place three cameras in an arbitrary arrangement that allows each one view the helicopter's flight volume. After an unplanned path, they obtain the camera's extrinsic parameters allowing a real-time positioning system get the helicopter's position in a camera array based coordinate system.

We also propose the use of an external vision system for estimating the helicopter's position but we just employ a single camera. At the current stage of research we have considered that the measurement of the depth, i.e. distance from the camera to the helicopter, can be suppressed, limiting the aircraft maneuvers to a plane parallel to the image. Taking the appropriate reference points we can easily generalize the position estimation to the 3D space [5].

2 Modeling the Flight Dynamics

Our main and final objective is to control the flight of a conventional radio-controlled helicopter by means of a computer using an external computer vision system. The outline of our approach is basically described by the tasks carried out by the three main subsystems: (1) the digital image capturing and processing subsystem, (2) the creation of computer-based RC transmitter, and (3) the flight control subsystem.

The digital image subsystem is in charge of the image capturing and all the processing needed to determine the position and orientation of the helicopter in a predefined coordinate frame. The next section is dedicated to describe the procedures and algorithms which allow us estimate the position information, thus, here we only enumerate the steps in which the whole process is divided.

The first step is the capture of the digital image of the scene in which the helicopter is. For this purpose we employ a conventional software package as Coriander (http://damien.douxchamps.net/ieee1394/coriander/) and Apple iSight digital cameras. They have a CCD color sensor and a FireWire 400 bus for communicating with the host. We are working with color images with a resolution of 640×480 pixels and a frame rate of 30 fps. The captured images are then segmented for detecting the zone in which the helicopter is. This step will be detailed in the next section. The last step is made by the tracking module. The goal of this module is to determine coherent relationships between the features of two consecutive images. The features can be corners, lines, or any element which can be tracked from an image to the next one and allows to show the movement between them. By means of the tracking, we can obtain speeds, future positions and orientations, etc. of the scene object.

Additionally there exists a calibration module that transforms the helicopter position and orientarion expressed in image coordinates to the position and orientation referred to an external arbitrary coordinate frame. The results presented in this paper do not include the use of this module. In other words, all the measurements are expressed in image coordinates.

In the sequel we will comment the most important aspects of some modules in detail. In particular we will describe how the helicopter position is determined, the way in which we acquire the data of the sticks commands sent to the helicopter, and how these informations are mixed to generate the datasets used for the neural network training.

3 Digital Image Processing

The helicopter's position in the image must be calculated in order to estimate the real 3D position. We have checked two methods to solve this task. Both are standard methods based on the detection of moving elements in the scene. As it is well known, these methods use and compare several scene images for determining which objects are in movement.

Initially we also considered another segmentation procedures, as for example color-based or markers-based, but one of our global goal is to avoid the adaptation or modification of the aircratf. Usually the models' payloads and masses are optimized and a small change in these values could create interferences and change the flight dynamics, achieving an unstable model. In fact, we did some initial tests in which we used markers made with small colored pieces of paper glued to one of the helicopter sides and they caused an incorrect level of the horizontal flight plane, so, the helicopter was lightly turned to that side.

One of the methods is the background substraction segmentation. Basically, this method proposed to compare each new frame to a reference image composed by all the static objects in the scene. This reference image is known as the *background* (see Fig. 1a). Thus, when a new frame arrives (Fig. 1b), it is compared to the background. Fig. 1c shows the output. When a pixel of the new image is different enough to the original background image, it is represented by a white dot in the output. The zones represented in black have similar values for the color intensities.



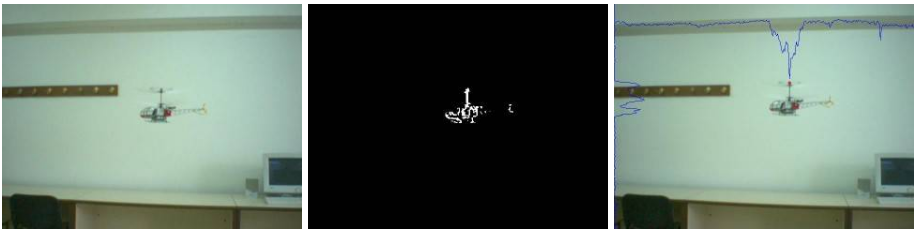
(a) (b) (c)

Fig. 1. Segmentation based on background

This method is very sensitive to changes in the light conditions and camera movements. In our case, an indoor environment, the camera movements can be easily avoided but the troubles by illumination must be controlled. We have

selected the image shown in Fig. 1c because it can be a good example of how these issues can modify the output: the bottom zone includes the table's edge and also the computer monitor which, obviously, are immobile objects. However, it can be easily fixed updating continuously the model of the background.

The second method studies the temporal differences between two or more consecutive frames in order to detect the zones in which there was any movement. In dynamic environments, the results can be better than the provided by other techniques because we are constantly comparing new images. However, it could not detect movement in pixels or zones of some objects. Fig. 2 shows the original frame (a) and the output (b). As in the previous method, white pixels are used to determine if a movement has been detected.



(a) (b) (c)

Fig. 2. Segmentation based on temporal differences and corner detection

For our purpose we use both methods although the background-based is more accurate in some situations. The temporal difference-based could generate some erroneous outputs due to its inherent characteristics.

Once the output is obtained we use conventional low level image processing techniques for estimating the aircraft position in the image. The Fig. 2c shows an horizontal and a vertical histograms that are used to get several reference points. In this case we are obtaining the point in the blades rotation axis. Other points can be easily obtained. The whole result of this subsystem gives one or more helicopter's reference points which will be used in combination to the transmitter sticks changes to estimate its movements. The next section shows how the stick data acquisition is made.

4 Stick Data Acquisition

4.1 Link

The helicopter selected for our experimental tests is the Hirobo Lama XRB-SR. With its contrarotating main blades, we get a very high stability without losing manoeuvrability.

From the standard helicopter configuration, we decided to use the lightweight stabilizer in order to reduce the resistance and weight; and regarding the manufacturer's original power supply based on Li-Poly batteries, we decided to improve

the system by assembling a different model, KOKAM, which goes from 720 mA to 940 mA with very similar Cs. Thus, we obtain more flying time.

This model is a four channel radio controlled helicopter. We use the Futaba SkysPort T4YF FM transmitter. This radio transmitter has four channels and a master-slave circuit used to create a link to other devices (mainly, other transmitters in master-slave configurations). In our case, we chose the 85153 adapter from Multiplex to connect the transmitter to the PC by using the USB port.

Once the transmitter-PC link is created, we are able to read and capture the stick movements during the manual (human) flight.

4.2 Sticks Definition

The radio transmitter has two sticks to control four channels and the association of them depends on the configuration and mode defined. The left stick controls the rudder (channels #2 and #4, vertical and horizontal, respectively) and the right one the throttle (channels #1 and #0, vertical and horizontal, respectively).

During the flight, the helicopter is affected by external conditions like air masses and perturbations created by the helicopter itself. Mechanical deviations, floor effect, inertia, battery level and multiple variables force the human pilot to control the two sticks concurrently in order to totally control the helicopter.

Per stick, the pilot counts with two trimmers in order to introduce off-set values associated to each vertical and horizontal movement. These fine adjustments let the pilot avoid some mechanical deviations at the trasmitter level.

We are interested in capturing the real stick values transmitted to the Helicopter during the flight. In the case that the radio is trimmered, these off-sets are added to the signal.

4.3 Calibrator Software

In order to connect the Skysport transmitter to a PC, we select the same configuration used by the flight simulator software using an USB adaptor. The link is simple and unidirectional from the transmitter to the PC, and once the connection is created, the transmitter is seen as a joystick peripheral.

From the computer side, we use a Linux box (no Real Time kernel needed) to deploy and run a capturator software capable to read the data provided by the USB interface using a simple pooling methodology.

Coded in Perl language and using the GTK library for the user interface layer, the calibrator software handles the standard joystick driver to process the information. We also introduce some nice features into the package looking for a chance to make a fine tuning on demand.

Besides this, Calibrator shows the value captured on each channel and also the number of samples captured which are in both cases, relevant pieces of information in order to monitor the right evolution of the data acquisition process.

Although Calibrator uses the pooling methodology to read the interface, only the variations are stored. At the driver level and for each capture, the data at

the four channels are compared with the last read and only when a variation is found, the new updated value is stored.

Using Calibrator, we have the opportunity to measure the Skysport transmitter quality, not only by finding the limits of each range associated to the sticks, but also by evaluating any jump into the potentiometers.

4.4 Data files

As Calibrator only stores variations, the size of the sticks information isn't huge at any time. This brings us the possibility to store it at memory level during the capture and only make a dump when the process is finished.

The Calibrator software provides up to three output data formats: plain text, XML and CSV. In this case we work with the plain text, where we can find the four channel values per line with their time stamp as follows:

```
Timestamp         #  CH-2  # CH-4  # CH-1  # CH-0
1120905771,92263 #  26687 # 00000 # 00000 # 00000
1120905772,11358 #  26349 # 00000 # 00000 # 00000
1120905772,17866 #  24322 # 00000 # 00000 # 00000
1120905772,33663 #  23647 # 00000 # 00000 # 00000
```

5 Modeling the Helicopter Dynamics

In order to obtain a model of the flight dynamics we propose the use of an artificial neural network (ANN). The ANN can estimate the next aircraft position being known the current position and the stick values. It is relevant that we do not use a specific position of the helicopter or the sticks, rather than we work with the variation between two consecutive positions. In other words, taking as inputs the helicopter speed in all the axes of the reference system and the variation of the sticks values, we want to estimate the future helicopter speed which allows us integrate the future position. This scheme is depicted in Fig. 3.

To create the sample sets we employ real values generated by means of the analysis of real image sequences and real stick values. Both, images an stick values, are obtained using the radio-controller connected to a computer running the image capture software and storing each image with the current sticks positions.

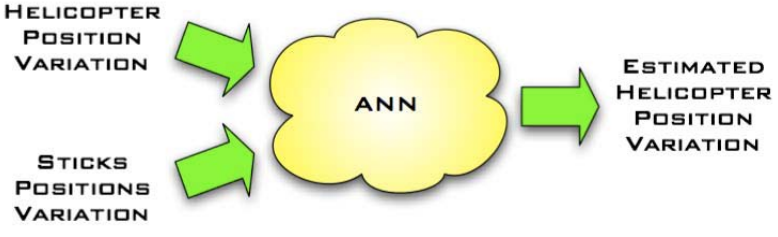


Fig. 3. Basic artificial neural network topology

Due to the fact that this system presents a high degree of inertia we must use recurrent neural networks [6]. Thus, the model of ANN can be interpreted in such a way that the input is not composed by a simple reference speed but it is a sequence of reference points.

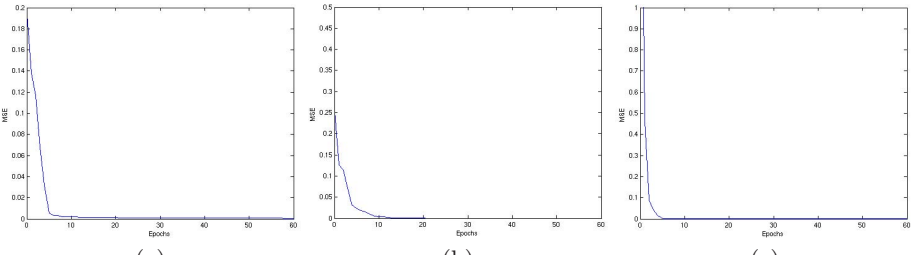
This kind of topology is implemented by means of multilayer perceptrons with a hidden layer and several delays in the input layer. We explain the impact of the use of the delay lines in the next section.

The training process was made using the Neural Network Toolbox of Matlab. During the training phase we used the Levenberg-Marquardt optimization algorithm. There were used from 4 to 64 delay lines and from 8 to 256 neurons in the hidden layer. Both the inputs and the outputs were normalized. Also, several learning sets were used. They differ in the sequence length, i.e. the number of samples, varying from 100 to 2000 pairs. Some sets are used for learning purposes and others for cross-validation.

6 Experimental Results

The tests were made with several training sets and defining several network topologies. The most simple cases only include one helicopter coordinate (the horizontal or the vertical coordinate referred to the image coordinate system) and the associated radio-controller channel. The inputs, and also the outputs, were progressively incremented until they reached the most complex network possible which has inputs for all the helicopter coordinates and for the four different stick axis. The tests also consider several helicopter orientations as for example a lateral view or a frontal view.

Fig. 4 shows some training curves. All of them present similar shapes. The best case (c) was obtained with an ANN with six inputs (horizontal and vertical helicopter coordinates and the four radio-controller channels), with 32 delay inputs and 32 neurons in the hidden layer. The MSE was 1.507510^{-31}. Depending on the run, it was necessary from 150 to 1000 epochs per training. If we only use two radio-controller channels, the case (b), rather than four (forward/backward and up/down) the results are quite similar: also 32 delays, 16 neurons for the hidden layer and a MSE of 1.242410^{-30}. The case (a) is the worst and simplest:



(a) (b) (c)

Fig. 4. Curves of the ANN learning process. The details are explained in the text.

the MSE was 3.089910^{-13}, with 32 delays and 32 neurons in the hidden layer, but just one channel and one coordinate (the forward/backward and the horizontal, respectively) were used.

7 Conclusions and Further Work

We have presented a framework that allows a bidirectional communication with a conventional RC transmitter by means of a mini computer board running GNU/Linux. The framework allows us obtain data from the RC controller and send commands to the aircraft. That information has been used in combination to sequences of images for generating sets of pairs which associate the helicopter movement and the command control produced with the controller sticks. These sets of pairs are used as training sets for artificial neural networks which are able to learn and generalize those relationships and provide a soft-computing model of the helicopter flight dynamics.

When the data sets for the training are small (over 100 samples), we need around half a dozen of delay inputs in the recurrent network for achieving low error values. An increment in the size of the sets implies an associated increment in the number of delays up to 32 per input.

We have checked that the increments in the number of delays over 32 do not improve the average error. Since we are using image sequences with 30 frames per second, the network is fed up, aproximately, every second. This fact possibly rebates our initial hypothesis and probably the helicopter is not as faster as we supposed.

Currently we are studying how the reduction of the frame rate can affect the model and its network topology. We are also improving some of the image processing and computer vision algorithms. We are specifically upgrading the background segmentation with an improved method to obtain the background model and also incorporating a new calibration module to the system.

References

1. Sugeno, M., Hirano, I., Nakamura, S., Kotsu, S.: Development of an intelligent unmanned helicopter. IEEE Int. Conf. on Fuzzy Systems 5, 33–34 (1995)
2. Corke, P., Sikka, P., Roberts, J.M.: Height Estimation for an Autonomous Helicopter. In: Int. Symposium on Experimental Robotics, pp. 101–110 (2000)
3. Roberts, J.M., Corke, P.I., Buskey, G.: Low-cost flight control system for a small autonomous helicopter. In: IEEE Int. Conf. on Robotics and Automation, vol. 1, pp. 546–551 (2003)
4. Matsuoka, M., et al.: Autonomous helicopter tracking and localization using a self-surveying camera array. In: Proc. 5th Int. Conf. on Field Service Robotics (2005)
5. Quan, L., Lan, Z.: Linear N-Point camera pose determination. IEEE Trans. on Pattern Analysis and Machine Intelligence 21(8), 774–780 (1999)
6. Elman, J.L.: Finding structure in time. Cognitive Science 14, 179–211 (1990)

Movement Identification Analysis Based on *Motion Capture*

Ryszard Klempous

Institute of Computer Engineering,
Control and Robotics, Wroclaw University of Technology
27 Wybrzeze Wyspianskiego Street, 50-370 Wroclaw, Poland
ryszard.klempous@pwr.wroc.pl

Abstract. This paper introduces motion capture a technique of digitally recording motion for entertainment, sports and medical applications. Presented analysis of possibilities as well as some modifications is an important part of a complex identification system. The results discussed in this paper can help to determine what kinds of methods could be best applied in a given context.

1 Introduction

Biometrics (ancient Greek: bios ="life", metron ="measure") is the study of methods for uniquely recognizing humans based upon one or more intrinsic physical or behavioral traits. In information technology, biometric authentication refers to technologies that measure and analyze human physical and behavioral characteristics for authentication purposes. It is a much more difficult task than verification because we have to compare those info with all the other entries of the database and the result must provide one clear answer [30]. Biometry according to the definition given by the Department of Biometrics at Cornell University is the application of statistics, probability, mathematics, systems analysis, operations research, engineering, computer science, and other areas to studies of the life sciences. Possible practical applications are not limited to following ones: access control and privileges checking (1), suspect identification (2), security systems (also personal tracking) (3), medical data analysis (diagnosis/treatment) (4). The emphasis is on aspects related to improving public safety as well as medical applications. Different methods of biometric identification of persons such as fingerprint analysis, otoscopy, retina matching, or voice analysis, new concepts are under intensive development. Systems come to existence that use behavioral biometry based on handwriting, patterns of using computer keyboard, traseology and motion analysis of various body parts.

1.1 Motions Capture Technique

Currently a motion capture technique [3,27,29] is very willingly used for creation of realistic human-like figures animations. There are two most often used types of

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 629–637, 2007.
© Springer-Verlag Berlin Heidelberg 2007

this technique. In the first case reflective markers are fixed on joints of a live actor and the motion of markers is tracked. In the latter case magnetic sensors are fixed on actor joints. These sensors are tracking disturbances of magnetic field during motion. In order to achieve realistic animation there is recorded motion of each human joint. This causes that it is necessary to describe motion with a large set of data. Such data are hard to process in some fields of applications. This problem is especially visible in use of multimedia databases. Managing the tremendous amounts of data is often supported by clustering and classification methods. It is not easy to find such methods for motion sequences. In our approach we try to solve this problem [19,25]. The main points of MC are:

1. Actor's motion is captured using different kinds of sensors and recorded in 3D virtual space.
2. Data are saved using proper representation taking into consideration skeleton hierarchy.
3. In this technique many subtle details can be recorded, which are difficult to simulate using analytical motion model.

In this paper we will concentrate on the results rather than on the process of capturing. Data obtained during capturing are usually highly noisy due to many factors like the relatively low resolution of capturing devices or algorithms used to pre-process data [5,6]. To be able to use the data, different kinds of de-noising methods have to be used [11,15]. Data that is used in the sequences are multidimensional and have built-in space-time dependency. Motion is captured in a real environment where all physical constraints take effect. Although tracking from each sensor on actor's body is obtained individually, all of them are connected with each other in the skeleton hierarchy. Therefore improper signal processing can result in violation of both the figure's structure and physical laws. The solution could be either to take into consideration all forces and real environmental constraints or to use methods which do not introduce skeleton structure modification [7]. The first approach is difficult to obtain because it is almost impossible to create a realistic animation in an analytical way, considering all real constraints. In the paper we will show that the second approach can be partially achieved by using correct data representation and properly prepared de-noising algorithms.

2 Motion Model of Human Gait

In this paper an attempt to use only quaternions discovered by William Hamilton [9,26] for coding the orientation of a human skeleton is proposed. Further transformations during motion processing are carried out in the quaternions domain using their specific properties. The approach is similar to concepts of Barr et al. [1], where rotations of a given object are converted to quaternions and later on an interpolation scheme is constructed. In [30] a method for motion fairing is presented, taking into consideration both rotational and translation motion

information. Energy criterion and minimization schemes are constructed in order to obtain a filtered time-series. The idea is generalized in [26] for multi-resolution analysis and successfully applied for motion blending.

2.1 Motion Model Description

The model of a human skeleton under considerations that consists of fixed number of bones (usually is between 18 and 23) organized in an invariable hierarchy. Each bone can have at most one predecessor and one or more successors. Each skeleton has its virtual root bone. The bones can change their positions only by the motion of their predecessors. Human bones can move only by rotation around given axes. Only information about the rotation of each bone is stored. Usually, only information about the rotation of each bone is recorded. Additionally, only one set of data representing global translation of the skeleton is defined and it is usually only for the root bone. There are two convenient types of data representations for rotations: Euler's angles (together with rotational matrices) and quaternions [9].

2.2 Data Representation

The natural way to represent data of human motion is to describe rotations of each bone. Rotations can be expressed as the vector of three variables where k denotes a bone in the skeleton and t denotes time moment (e.g. frame number) in the sequence:

$$\bar{r_k}(t) = [r_{sk}(t), r_{yk}(t), r_{zk}(t)]^T , \tag{1}$$

The rotation is described as a rotation around three axes independently and is called Euler's rotations. The simplification of the approach is to analyze motion, which is performed into the perpendicular direction to the camera. The simplification was proposed in [2,29,30] and we base our experiments on that approach. The novelty presented here is using similar approach both for frequency as well as for time-series analysis of motion capture data. One of the most convenient alternative representations for rotations is a unit quaternion. The most important feature is that the unit quaternion can represent rotations [22]. Single quaternion is represented as:

$$q = w + x_i + y_j + z_k = [w\ x\ y\ z] = (s,\ v) \tag{2}$$

where w (or s) is a scalar part and v denotes vector part. Representation which is used for coding rotations is given as:

$$\bar{q} = (s, \overrightarrow{v}) = \left(\cos\frac{\theta}{2}, \overrightarrow{u}\sin\frac{\theta}{2}\right), \tag{3}$$

where u is a singular vector along which a rotation of angle θ is defined.

The measure of similarity between rotations takes into consideration scalar part of the quaternion [22]. The motion model is shortly presented on Figure 1.

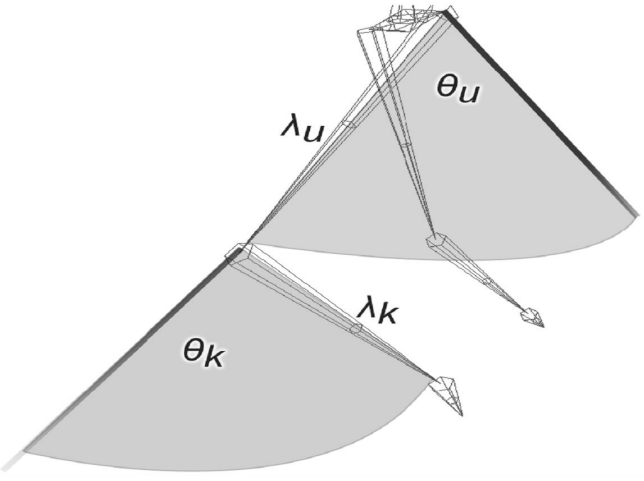


Fig. 1. The modified motion model of limbs [13]

Rotations are measured against the initial position of bones for a skeleton. It is easy to observe that a unit quaternion can represent in three-dimensional space rotation around an axis [9]:

$$q^t = (\cos\theta + \bar{u}\sin\theta)^t \tag{4}$$

where u is a unit vector defining the axis and θ is a rotation angle around it.

Expression (4) leads to the generalization of Euler's identity for complex numbers:

$$\exp(\bar{u}\theta) = \cos\theta + \bar{u}\sin\theta. \tag{5}$$

There are different kinds of quaternion interpolation. To find an interpolation between two quaternions Q10 and Q20 in the point $0 \leq p \leq 1$ one has to compute:

$$slerp(p; Q_{10}, Q_{20}) = Q_{10}\exp\left(p\log\left(Q_{10}^{-1}Q_{20}\right)\right) = Q_{10}\left(Q_{10}^{-1}Q_{20}\right)^p \tag{6}$$

Thanks to introduced above expressions, the result Qx0 of the interpolation is

$$Q_{x0} = slerp(p; Q_{10}, Q_{20}) = \frac{Q_{10}\sin((1-p)\,\phi) + Q_{20}\sin(p\phi)}{\sin\phi} \tag{7}$$

where ϕ is the acute angle between two initial quaternions.

Similarly, the following equivalent of the measure of distance:

$$Q_{21} = Q_{10}^{*}Q_{20} \tag{8}$$

where * denotes the conjugation of the quaternion. The operation (8) simply defines a quaternion, which is necessary to make a rotation between positions

defined by Q10 and Q20. It is worth mentioning that a single unit quaternion always using rotational interpretation represents a rotation between two positions. That is why in all expressions quaternions are marked with double subscripts: qij denotes a rotation of an object between reference frame the final frame j and i. We name them frames because of their application in animation systems. If we consider unit quaternions, which represent rotation, a good equivalent of the measure is:

$$d\left(Q_{10}, Q_{20}\right) = \arccos\left(Re\left(Q_{21}\right)\right) = \arccos\left(Re\left(Q_{10}^{*} Q_{20}\right)\right) \tag{9}$$

We investigated some properties of this measure for the application in motion data processing [16]. In the comparison with other representations the matrix K provides the full description of the motion. We assume that the skeleton consists of K bones (in this researches K was equal to 23). The animation length is n and depends on the actual sequence. Each cell of the matrix includes a unit quaternion representing rotation between the initial position of the bone k and the position in frame t.

$$K = \begin{bmatrix} q_1(1) & q_1(2) & \dots & q_1(n) \\ q_1(1) & q_2(2) & \dots & q_2(n) \\ \vdots & \vdots & q_k(t) & \vdots \\ q_K(1) & q_K(2) & \dots & q_K(n) \end{bmatrix} \tag{10}$$

Additionally the full motion model includes information about a structure of the skeleton.

3 Gait Recognition Task

The motion is the object, which is taken into account. The basic approach is to treat motion as a multi dimensional time-series of rotations of each bone in the skeleton. Murray [26,27] suggested that the motion of hips and legs differs significantly among different people. In the papers [10,12,13,23] rotations of shanks, hips and the neck are considered. The approach has similar problems as the previous – those parts of the body are usually hidden under the clothes. On the contrary another part, which could be analyzed, are legs. The approach is presented in papers [6,7,24,30]. The main advantage is the possibility to determine the position of these parts of the body also on the image. The motion is more periodic, which is not interfered with other behaviors like carrying or gesticulations. Taking into consideration those conditions, we decided to examine the motion of legs, especially two bones: hips and shanks. In all experiments presented in this paper we use LifeForms [24] motion library, which contains different action sequences played by real actors and processed using motion capture technique. In different researches it was presented that person can be identified using legs motion (Cunado D. at al [7]). The method can be successful for video analysis, where other parts of the body are covered by clothes – legs are easier

to distinguish (e.g. using Hougha transform: Cunado et al [8]. In the cited papers it is proposed to take into consideration rotation along one axis and only 3 coefficients of amplitude spectrum analysis (I, II amplitudes of X rotation of hip, I amplitude of X rotation of knee). Proposed method of extended motion model gives very interesting results. Application of filtering in motion sequences (equivalent of Gaussian blur for quaternions) results in significant improvement of clustering quality [12,18].

3.1 Related Works Concerning Biometric Motion Analysis

It should be extraction of biometric information from a person and the comparison them with gait database. We have to compare the data with the database and the result must be unique [20]. Two methods are proposed to recognize gaits. First, the Dynamic Time Warping (DTW) method is considered to compare motion sequences and second one is based on Spectrum analysis. The following legs data are taken into consideration: legs angle, legs length, steps frequency, foots positioning techniques, roots movement, hips movement.

3.2 The Dynamic Time Warping (DTW) Method

The method is based on dynamic programming and is widely used for different time-series comparison applications (i.e. voice recognition [9,28]). The application for motion processing was presented in different papers [23,24]. The proposal of database structure based on DTW was proposed in [1]. Two experiments were prepared. In each we selected one motion capture sequence called armout.bvh as a template sequence. Each motion from the group of sequences was compared to the template. The warping cost represents a measure of similarity between sequences. The properties of comparison algorithm allow to process signals with different lengths. The results are presented in [12,13,19] Warping costs for different rotations are placed on different axes. This method of data visualization is based on paper [30]. The novelty is to use it for DTW analysis of motion capture data. The closer points are placed on the graph, the more similar are analyzed sequences. Similar experiment was carried out using different parameters. This time we analyzed only X rotations, but taking into consideration different parts of the body: left hip, right hip and left shank. Due to the asymmetrical nature of such comparison. The results are presented in the [14,18,19].

3.3 Spectrum Analysis

The spectrum analysis of the motion signal can lead to interesting results concerning person identification [3,7,25]. Spectrum of the motion signal can be computed using Fourier transform.

Computed in this way spectrum can be afterwards analyzed and be the source of interesting biometric information. Kuan [17] showed that the spectrum parameters differ significantly for different persons. In [7,30] the approach was

extended with the application of phase-weighted magnitude. The similar data visualization are used in order to compare the results with other authors. Both spectrums were computed for rotational signals along X axis of the right hip. As one can observe, the result of spectrum analysis differs significantly for two sequences, which differs only with some details (Sprint and Walk represent motions of the same type) [12,19].

3.4 Experiment: Multidimensional Spectrum Analysis

The most difficult is to distinguish noised group of motions [10,11]. Also some elements of other groups do not form a clearly defined cluster. It is possible to indirectly check the performance of the comparison method. The idea is to run clustering algorithm and to check how well clustered groups we can obtain. Generated groups of amplitude coefficients can give much better results. In most cases it was enough to use 5 coefficients to obtain satisfactory results. The next step is to find a method, which could deal also with very similar motions like RunToWalk and JogToWalk. Those motions are so similar that even taking into account many amplitude coefficients would not give satisfactory recognition. One of the ideas is to use combined time-spectrum analysis. It could join the result obtained from Dynamic Time Warping comparison and spectrum analysis.

4 Conclusions

There are many attempts, which try to obtain algorithms that could identify the person on the image and extract motion from the stream. It will be probably possible in the years to come; especially that processing power of common processor is still increasing. The main advantage of MC [1] systems is the accuracy of recorded data. Captured motion contains a lot of subtle details because its source is the real actor. Motion can be subsequently processed and used in animation system to render a completely new figure. The technique is widely used in many movies, which require the creation of special characters. Motion capturing can also be very helpful for medical analysis (e.g. for examinations of bones system disorders). There are some trials of the application of the MC system for runners training. Biometric motion identification is under intensive development. Presented analysis of possibilities and some modifications is the base of much more complex identification system. Such system always requires well-defined human motion model, comparison algorithms and properly constructed motion database. The results obtained from the system will greatly depend on analysis methods.

References

1. Barr A., Currin B., Gabriel S. and J. Houghes, Smooth interpolation of orientations with angular velocity constraints, Computer Graphics, Vol. 26, 1992.
2. Bodenheimer B, Chuck Rose, Seth Rosenthal, and John Pella, "The Process of Motion Capture: Dealing with the Data," 1997.

3. Bruderlin A., Williams L., Motion Signal Processing, Computer Graphics, 29, 97-104,1995.
4. Chaczko Z., "NIE Models for Emergency Services Interoperability ". To be published in Proceedings of Computer Aided Systems Theory: EuroCAST 2007. Springer Lecture Notes in Computer Science, 2007.
5. Chaczko Z., Sinha S., "Strategies of Teaching Software Analysis and Design – Interactive Digital Television Games", ITHET 2006, Sydney, Australia, July 2006.
6. Cunado D. at al, Automatic extraction and description of human gait models for recognition purposes, Computer Vision and Image Understanding, 90(1), 2003.
7. Cunado D., Nixon M.S., Carter J.N., Using Gait as a Biometric, via Phase-Weighted Magnitude Spectra, Proc. of 1st Int. Conf. on Audioand Video-Based Biometric Person Authentication, 2003.
8. Eberly D., Quaternion Algebra and Calculus, Magic Software, http://www.magicsoftware.com/Documentation/Quaternions.pdf, 2002.
9. Itakura F., Minimum Prediction Residual Principle Applied to Speech Recognition, IEEE Trans. on Acoustics Speech and Signal Processing, AS23.1, 67-72, 1975.
10. Jabłoński B., Porównywanie generatorów ruchu animowanych postaci ludzkich, Diversitas Cybernetica in, Lecture Notes in Communication Series, (Ed. R. Klempous), Warsaw, 2005.
11. Jabłoński B., Klempous R., Kulbacki M., PDE-based filtering of motion sequences, Journal of Comput. and Applied Mathematics, vol. 189, no. 1/2, s. 660-675, Elsevier, 2006.
12. Jabłoński B., Klempous R., Majchrzak D., Models and methods for biometric motion identification, Annales UMCS, vol. 4, Lublin, 2006.
13. Jabłoński B., Klempous R., Majchrzak D., Feasibility analysis of human motion identification using motion capture, Proc. of the Int. Conference on Modelling, Identification and Control, Acta Press, 2006.
14. Jabłoński B., Kulbacki M., Klempous R., Segen J.,Methods for comparison of animated motion generators, Proc. of IEEE International Conference on Computational Cybernetics, Hungary, 2003.
15. Lee, J., Shin, S.Y.: Motion Fairing. In: Proc. of Computer Animation, Geneva, Switzerland, pp. 136–143 (1996)
16. Jablonski B., Kulbacki M., Klempous R. and J. Segen, Methods for Comparison of Animated Motion Generators, in: Proc. ICCC 2003, Siofok, Hungary, 2003.
17. Kuan E. L., Investigating gait as a biometric, Technical Report, Departament of Electronics and Computer Science, University of Sothampton, (1995).
18. Kulbacki M., Bak A., Segen J., "Unsupervised Learning Motion Models Using Dynamic Time Warping," Intelligent Information Systems 2002, pp. 217-226, Sopot, Poland, June 3-6, 2002.
19. Kulbacki M., Jablonski B., Klempous R., SegenJ., Learning from Examples and Comparing Models of Human Motion, Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 8, No.5, Fuji Technology Press 2004, 477-481.
20. Lee J.and. Shin S. Y, Multiresolution Motion Analysis with Applications, The international workshop and Human Modeling and Animation, Seoul, 2000, pp. 131-143.
21. LifeForms ,Credo Interactive Inc. http://www.credo-interactive.com/
22. Maillot P., Using quaternions for coding 3d transformations. In Graphics Gems I, Academic Press Inc., Boston, 498-515, 1990.

23. Majchrzak D., Identyfikacja ruchu postaci. Analiza możliwości, metody, algorytmy",M.Sc.Eng. Thesis, (in polish) Wroclaw University of Technology, Faculty of Electronics, 2000.
24. Myers C.S, Rabiner L. R., A comparative study of several dynamic time-warping algorithms for connected word recognition, The Bell System Technical Journal, 60(7), 1389-1409, 1981.
25. Mowbray S.D, Nixon M.S, Automatic Gait Recognition via Fourier Descriptors of Deformable Objects, Proc. of Audio Visual BiometricPerson Authentication, 2003.
26. Murray M. P., Gait as a total pattern of movement, Amer. J. Phys. Med. 46(1), 1967.
27. Murray M. P., Drought A.B., Kory R.C., Walking patterns of normal men, J. Bone Joint Surg. 46-A (2), 335-360, 1996.
28. Rabiner L.R.,. Juang B, Fundamentals of speech recognition, Englewood Cliffs, NJ: Prentice-Hall, 1993.
29. Yam C. Y., Automated person recognition by walking and running via model-based approaches, Pattern Recognition 37(5), Elsevier Science, 2003.
30. Yam C. Y., Nixon M. S., Carter J. N. Extended Model-Based Automatic Gait Recognition of Walking and Running. In Proc. of 3rd Int. Conf. on Audio- and Video-Based Biometric Person Authentication, pp. 278-283, AVBPA 2001.

Algorithmic Lateral Inhibition Formal Model for Real-Time Motion Detection

María T. López[1], Antonio Fernández-Caballero[1], Miguel A. Fernández[1], José Mira[2], and Ana E. Delgado[2]

[1] Universidad de Castilla-La Mancha
Instituto de Investigación en Informática (I3A) and
Escuela Politécnica Superior de Albacete, 02071 - Albacete, Spain
{mlopez,caballer,miki}@dsi.uclm.es
[2] Universidad Nacional de Educación a Distancia
E.T.S.I. Informática, 28040 - Madrid, Spain
{jmira,adelgado}@dia.uned.es

Abstract. Recently, the use of the algorithmic lateral inhibition (ALI) method in motion detection has shown to be very effective. The promising results in terms of the goodness of the silhouettes detected and tracked along video sequences lead us to accept the challenge of searching for a real-time implementation of the algorithms. This paper introduces two steps towards that direction: (a) A simplification of the general ALI method is performed by formally transforming it into a finite state machine. (b) A hardware implementation of such a designed ALI module, as well as an 8x8 ALI module, has been tested on several video sequences, providing promising performance results.

1 Introduction

In recent years, many researchers have explored the relation between discrete-time recurrent neural networks and finite state machines, either by showing their computational equivalence or by training them to perform as finite state recognizers from example [1]. The relationship between discrete-time recurrent neural networks and finite state machines has very deep roots [2]. The early paper mentioned show the equivalence of these neural networks with threshold linear units, having step-like transfer functions, and some classes of finite state machines. More recently, some researchers have studied the close relationships more in detail [3], as well as the combination of connectionist and finite state models into hybrid techniques [4].

An important issue in the motivation of this paper is that the performance of neural-based methods can be enhanced by encoding a priori knowledge about the problem directly into the networks [5]. This knowledge can be encoded into a recurrent neural network by means of finite state automata rules [6]. The second idea introduced is that such a finite state machine, implemented in hardware, may provide real-time performance. The algorithmic lateral inhibition (ALI)

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 638–645, 2007.
© Springer-Verlag Berlin Heidelberg 2007

method is precisely inspired in the (recurrent and non-recurrent) neural computation mechanism known as lateral inhibition. Our experience up to date has shown that most applications in computer vision, and more concretely in motion detection through the ALI method (ALI), offer good results [7]. And, currently our research team is involved in implementing the method into real-time in order to provide efficient response time in visual surveillance applications [8],[9]. This article shows how to implement the ALI method in motion detection by means of a formal model described as finite state machines, leading to an ALI module, and its further implementation in a programmable logic device, such as an FPGA.

2 Formal Model for ALI in Motion Detection

2.1 ALI Temporal Motion Detecting

The aim of this subtask is to detect the temporal and local (pixel to pixel) contrasts of pairs of consecutive binarised images at gray level k. The step firstly gets as input data the values of the 256 gray level input pixels $I(i,j;t)$ and generates $N = 8$ binary images, $x_k(i,j;t)$, corresponding to N levels defined by "bands". The output space has a FIFO memory structure with two levels, one for the current value and another one for the previous instant value. Thus, for N bands, there are $2N = 16$ binary values for each input pixel; at each band there is the current value $x_k(i,j;t)$ and the previous value $x_k(i,j;t-\Delta t)$, such that:

$$x_k(i,j;t) = \begin{cases} 1, \text{ if } I(i,j;t) \in [32 \cdot k, 32 \cdot (k+1) - 1] \\ 0, \text{ otherwise} \end{cases} \tag{1}$$

where $k = 0, 1, ..., 7$, is the band index. Thus, we are in front of a vector quantization (scalar quantization) algorithm generally called multilevel thresholding. As well as segmentation in two gray level bands is a usual thing, here we are in front of a refinement to the segmentation in N gray level bands. Thus, multilevel thresholding is a process that segments a gray-level image into several distinct regions.

Now, each computation element at this stage, $y_k(i,j;t)$, gets a charge value, complemented by label A_C, a binary signat that is also updated, according to the following formulas:

$$A_C = \begin{cases} 1, \text{ if } (x_k(i,j;t) = 1) \cap (x_k(i,j;t-\Delta t) = 0) \\ 0, \text{ otherwise} \end{cases} \tag{2}$$

$$y_k(i,j;t) = \begin{cases} v_{dis}, \text{ otherwise} \\ v_{sat}, \text{ if } (x_k(i,j;t) = 1) \cap (x_k(i,j;t-\Delta t) = 0) \\ \max[x_k(i,j;t-\Delta t) - v_{dm}, v_{dis}], \\ \qquad \text{if } (x_k(i,j;t) = 1) \cap (x_k(i,j;t-\Delta t) = 1) \end{cases} \tag{3}$$

2.2 ALI Spatial-Temporal Recharging

In the previous step ALI Temporal Motion Detecting we have obtained the individual "opinion" of each computation element. But, our aim is also to consider the "opinions" of the neighbors. The reason is that an element individually should stop paying attention to motion detected in the past, but before making that decision there has to be a communication in form of lateral inhibition with its neighbors to see if any of them is in state S_7 (v_{sat}, maximum charge). Otherwise, it will be discharging down to S_0 (v_{dis}, minimum charge), because that pixel is not bound to a pixel that has detected motion. In other words, the aim of this step is to focus on those pixels charged with an intermediate accumulated charge value, $y_k(i,j;t)$, but directly or indirectly connected to saturated pixels (v_{sat}) in state S_7 by incrementing their charge. These "motion values" of the previous layer constitute the input space, whereas the output is formed after dialogue processing with neighboring pixels by the so called permanency value, $z_k(i,j;t)$.

The values of charge accumulated before dialogue are written in the central part of the output space of each pixel (C^*) that now enters in the dialogue phase according to recurrent ALI. The data in the periphery of receptive field in the output space of each pixel (P^*) contains now the individual calculi of the neighbors. Let $v_C(t) = y_k(i,j;t)$ be the initial charge value at this step. Each pixel takes into account the set of individual calculus, $v_C(t + k \cdot \Delta\tau), A_j$, according to:

$$A_{P^*}(\tau) = \bigcup_j A_j(\tau) \tag{4}$$

This result, A_{P^*}, is now compared with A_C, giving rise to one of two discrepancy classes (recharge or stand-by).

$$D(t + l \cdot \Delta\tau) = \begin{cases} stand - by(v_{dis}), \text{ if } v_C(t + l \cdot \Delta\tau) = v_{dis} \\ stand - by(v_{sat}), \text{ if } v_C(t + l \cdot \Delta\tau) = v_{sat} \\ recharge, \text{ if } (v_{dis} < v_C(t + l \cdot \Delta\tau) < v_{sat}) \cap (A_{P^*} = 1) \end{cases} \tag{5}$$

Subsequently, the class activated outputs the new consensus charge value after dialogue, $z_k(i,j;t + \Delta t)$, with $\Delta t = k \cdot \Delta\tau$, being k the number of iterations in the dialogue phase, a function of the size of the receptive field. Notice that τ is a parameter that only depends on the size of the objects we want to detect from their motion. So, the purpose of this inference is to fix a minimum object size in each gray level band. The whole dialogue process is executed with clock τ, during k intervals $\Delta\tau$. It starts when clock t detects the configuration $y_k(i,j;t - \Delta t) = y_k(i,j;t) = 1$ and ends at the end of t, when a new image appears.

$$A_C = \begin{cases} 1, \text{ if } D(t + l \cdot \Delta\tau) = \{stand - by(v_{sat}) \cup recharge\} \\ 0, \text{ otherwise} \end{cases} \tag{6}$$

$$v(t+l\cdot\Delta\tau)=\begin{cases}v_{dis},\ \text{if } D(t+l\cdot\Delta\tau)=stand-by(v_{dis})\\ v_{sat},\ \text{if } D(t+l\cdot\Delta\tau)=stand-by(v_{sat})\\ \min[v(t+(l-1)\cdot\Delta\tau)+v_{rv},v_{sat}],\\ \qquad\qquad\text{if } (D(t+l\cdot\Delta\tau)=recharge\end{cases}\tag{7}$$

$$A_C=0,\ \text{if } D(t+(l-1)\cdot\Delta\tau)=\{stand-by(v_{sat})\cup recharge\}\tag{8}$$

In each dialogue phase (in other words, in each interval of clock $\Delta\tau$), the calculation element only takes into account values $y_k(i,j;t-\Delta t)$, $y_k(i,j;t)$ and $A_C(t)$ present in that moment in its receptive field. To diffuse or to use more distant information, new dialogue phases are necessary. That is to say, new inhibitions in $l\cdot\Delta\tau$ $(1<l\leq k)$ are required. This only affects to state variable $A_C(\tau)$, as $y_k(i,j;t-\Delta t)$ and $y_k(i,j;t)$ values remain constant during the intervals used to diffuse τ and to consensus the different partial results obtained by the calculation elements.

Notice that the recharge may only be performed once during the whole dialogue phase. That is why $A_C=0$, when a recharge takes place. Lastly, the output will be:

$$z_k(i,j;t+\Delta t)=v_C(t+\Delta t)\tag{9}$$

Charge values, $z_k(i,j;t+\Delta t)$, offered by the previous step are now evaluated in the center and in the periphery. Now, let v_C be the initial charge value at this subtask. In P^* we have the average of those neighbors that have charge values different from θ_{min}, the so called "permanency threshold value".

$$v_C=\max[v_C,\theta_{min}]\tag{10}$$

Now the result of the individual value (C) is compared with the mean value in (P) and produces a discrepancy class according with threshold, θ_{min}, and passes the mean charge values that overcome that threshold. After this, the result is again compared with a second threshold, namely θ_{max}, eliminating noisy pixels pertaining to non-moving objects.

$$O_k(i,j;t+\Delta t)=\begin{cases}\theta_{min},\ \text{if } v_C=\theta_{min}\\ (v_C+v_P)/2,\\ \ \text{if } (\theta_{min}<v_C<v_{sat})\cap(\theta_{min}<v_P<v_{sat})\\ v_C,\ \text{if } (\theta_{min}<v_C<v_{sat})\cap(v_P=\theta_{min})\end{cases}\tag{11}$$

$$O_k(i,j;t+\Delta t)=v_{dis},\ \text{if } O_k(i,j;t+\Delta t)>\theta_{max}\tag{12}$$

The transitions among the initial state $S_i(t)$ (whenever $S_i(t)$ different from S_0) and the final state $S_i(t+\Delta t)$ are carried out in agreement with rule:

$$S_{i_{final}}=1/N_{k+1}(S_{i_{initial}}+\sum_{RF_k}v_j)\tag{13}$$

where the sum on sub-index j extends to all neighbors, v_j, belonging to the subset of the receptive field, RF_k, such that its state is different from S_0, and N_k is the number of neighbors with state different from S_0.

3 Real-Time Hardware Implementation of Motion-Detection ALI Modules

In order to accelerate their performance, and hence to obtain real-time processing rates, many applications use reconfigurable hardware. More concretely, they are programmed on field programmable gate arrays (FPGAs) [10]. Some of the most recently used FPGA families are Xilinx Virtex-II [11],[12],[13] and Virtex-E [14],[15].

In this section, we show how a single ALI module, as well as its expansion to an $8 * 8$ module, starting from the formal description as finite state machines has been implemented (see figure 1). In order to implement the module, the programming has been performed under Very High Speed Integrated Circuit Hardware Description Language (VHDL), and by means of the Xilink ISE 8.1 tool, the module has been synthesized and implemented in a Xilink Virtex-4 FPGA. More concretely, the device used is a 4vlx25ff668-10.

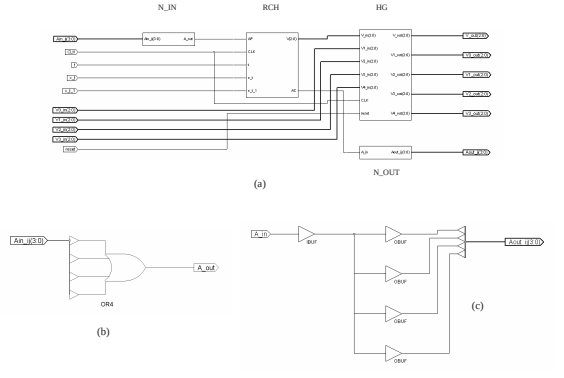


Fig. 1. (a) Layout of a motion-detection ALI module, N_IN is the input from the neighbors sub-module, N_OUT is the output towards the neighbors sub-module, RCH is the recharge sub-module, and HG is the homogenization sub-module. (b) Inside the N_IN module. (c) Inside the N_OUT module.

In Table 1, the temporal results associated to the implementation are shown, and in Table 2, the necessary logic for the implementation is offered. Now, for the implementation of an $8 * 8$ module, using the same FPGA (the 4vlx25ff668-10), the results obtained are shown in Tables 3 and 4.

The most relevant data is that clock CLK (τ in our formal model) can work at a frequency of 7.730 MhZ. Nevertheless, real results will be obtained at a higher time scale (t). When working with $8*8$ modules, the CLK highest frequency has

Table 1. Temporal results for the ALI module

Minimum period	10.916ns
Maximum frequency	91.609MHz
Maximum combinational path delay	12.357ns
Minimum input required time before clock	6.629ns
Maximum output delay after clock	14.672ns

Table 2. Logic distribution for the ALI module

Number of occupied Slices	43 out of 10,752 (1%)
Number of bonded IOBs	40 out of 448 (8%)
Number of BUFG/BUFGCTRLs	1 out of 32 (3%)
Total equivalent gate count for design	634

to be divided by 8. That is to say, the results for $8*8$ modules will be obtained at a frequency of 0.966 MHz (1.035 μs). When working with $512*512$ pixel images, which need 4096 $8*8$ ALI modules, the results are obtained after 4.24 ms. This result may be considered as excellent, as in order to work in real-time we have up to 33 ms per image frame.

4 Data and Results

In order to test the validity of our implementation, in this section the result of applying $8*8$ ALI modules on specific areas of a well-known benchmark image sequence is shown. Figure 2 shows the first and last images of the famous Hamburg Taxi scene, where we have drawn the $128*64$-pixel zone tested.



Fig. 2. Hamburg Taxi sequence. (a) Frame number 1. (b) Frame number 19.

Figure 3 shows the result on frames 5 and 19 of the sequence. As expected, due to the region growing technique underlying the ALI method, the silhouette of the car slightly goes appearing. As you my appreciate, in frame #5 only a little portion of the moving car's silhouette appears. This is because no motion has been detected in a great part of the car respect to the initial frame. Nevertheless,

Table 3. Temporal results for the ALI $8 * 8$ module

Minimum period	129.367ns
Maximum frequency	7.730MHz
Minimum input required time before clock	10.147ns
Maximum output delay after clock	5.829ns

Table 4. Logic distribution for the ALI $8 * 8$ module

Number of occupied Slices	3,097 out of 10,752 (28%)
Number of bonded IOBs	131 out of 448 (29%)
Number of BUFG/BUFGCTRLs	2 out of 32 (6%)
Number used as BUFGs	2
Total equivalent gate count for design	47,292



Fig. 3. Results on Hamburg Taxi sequence. (a) Result after frame number 5. (b) Result after frame number 19.

in frame #19 mostly the complete silhouette of the car may be observed, as at this frame enough motion exists respect to the initial frame.

5 Conclusions

The design by means of programmable logic enables the systematic and efficient crossing from the descriptions of the functional specifications of a sequential system to the equivalent formal description in terms of a Q-states finite state automata or a N-recurrent-neurons neuronal network, where $Q \leq 2^N$. Starting from this point, a hardware implementation by means of programmable logic is very easy to perform. This kind of design is especially interesting in those application domains where the response time is crucial (e.g. monitoring and diagnosing tasks in visual surveillance and security).

In this paper, the results obtained after implementing ALI modules in hardware on programmable logic, concretely on Virtex-4 FPGA's, have been shown. These results start from previous validated researches on moving objects detection, which unfortunately did not reach real-time performance. Prior to the implementation, a simplification of the model into an 8-state finite automaton has been performed. The procedure is easily expandable to all delimited-complexity functions that may be described in a clear and precise manner by a not too high number of states, which alternatively are capable of getting the module of the function.

Acknowledgments

This work is supported in part by the Spanish CICYT TIN2004-07661-C02-01 and TIN2004-07661-C02-02 grants, and the Junta de Comunidades de Castilla-La Mancha PBI06-0099 grant.

References

1. R.P. Ñeco, and M.L. Forcada, Asynchronous translations with recurrent neural nets, in Proceedings of the International Conference on Neural Networks, ICNN'97, vol. 4, pp. 2535-2540, 1997.
2. W.S. McCulloch, and W.H. Pitts, A logical calculus of the ideas immanent in nervous activity, Bulletin of Mathematical Biophysics, vol. 5, pp. 115-133, 1943.
3. R.C. Carrasco, and M.L. Forcada, Finite state computation in analog neural networks: Steps towards biologically plausible models, in Lecture Notes in Computer Science, vol. 2036, pp. 482-486, 2001.
4. F. Prat, F. Casacuberta, and M.J. Castro, Machine translation with grammar association: Combining neural networks and finite state models, in Proceedings of the Second Workshop on Natural Language Processing and Neural Networks, pp. 53-60, 2001.
5. J. Shavlik, Combining symbolic and neural learning, Machine Learning, vol. 14, no. 3, pp. 321-331, 1994.
6. C.W. Omlin, and C.L. Giles, Constructing deterministic finite state automata in recurrent neural networks, Journal of the ACM, vol. 43, no. 6, pp. 937-972, 1996.
7. J. Mira, A.E. Delgado, A. Fernández-Caballero, and M.A. Fernández, Knowledge modelling for the motion detection task: The lateral inhibition method, Expert Systems with Applications, vol. 7, no. 2, pp. 169-185, 2004.
8. M.T. López, A. Fernández-Caballero, M.A. Fernández, J. Mira, A.E. Delgado, Visual surveillance by dynamic visual attention method, Pattern Recognition, vol. 39, no, 11, pp. 2194-2211, 2006.
9. M.T. López, A. Fernández-Caballero, M.A. Fernández, J. Mira, A.E. Delgado, Motion features to enhance scene segmentation in active visual attention, Pattern Recognition Letters, vol. 27, no. 5, pp. 469-478, 2006.
10. F. Bensaali, and A. Amira, Accelerating colour space conversion on reconfigurable hardware, Image and Vision Computing, vol. 23, pp. 935-942, 2005.
11. I. Amer, W. Badawy, and G. Jullien, A proposed hardware reference model for spatial transformation and quantization in H.264, Journal of Visual Communication and Image Representation, vol. 17, pp. 533-552, 2006.
12. H. Moon, and R. Sedaghat, FPGA-based adaptive digital predistortion for radio-over-fiber links, Micropocessors and Microsystems, vol. 30, pp. 145-154, 2006.
13. S. Bojanis, G. Caffarena, S. Petrovic, and O. Nieto-Taladriz, FPGA for pseudorandom generator cryptanalysis, Micropocessors and Microsystems, vol. 30, pp. 63-71, 2006.
14. I.W. Damaj, Parallel algorithms development for programmable logic device, Advances in Engineering Software, vol. 37, no. 9, pp. 561-582, 2006.
15. S. Perri, M. Lanuzza, P. Corsonello, and G. Cocorullo, A high-performance fully reconfigurable FPGA-based 2-D convolution processor, Micropocessors and Microsystems, vol. 29, pp. 381-391, 2005.

Second Order Variational Optic Flow Estimation

L. Alvarez, C.A. Castaño, M. García, K. Krissian, L. Mazorra, A. Salgado, and J. Sánchez

Departamento de Informática y Sistemas, Universidad de Las Palmas de Gran Canaria, Campus de Tafira s/n, 35017 Las Palmas de Gran Canaria, Spain
Tel.: +34 928 458708, Fax: +34 928 458711
{lalvarez,ccastano,mgarcia,krissian,lmazorra,asalgado, jsanchez}@dis.ulpgc.es

Abstract. In this paper we present a variational approach to accurately estimate the motion vector field in a image sequence introducing a second order Taylor expansion of the flow in the energy function to be minimized. This feature allows us to simultaneously obtain, in addition, an estimation of the partial derivatives of the motion vector field. The performance of our approach is illustrated with the estimation of the displacement vector field on the well known Yosemite sequence and compared to other techniques from the state of the art.

1 Introduction

Optic flow estimation is a problem that has focused the attention of many researchers in the domain of image processing, probably due to the huge amount of applications in which the estimated displacement between consecutive frames is an important source of information. In this sense, many different approaches have been presented in the literature, starting from classical methods like the ones proposed by Horn and Schunck [1] or by Lucas and Kanade [2], trying either to overcome some limitations of the existing methods or to exploit some *a priori* information in order to improve the accuracy on the estimation. For instance, different regularization terms have been proposed in variational approaches like in [3,4], pyramidal decompositions have also been proposed in order to detect large displacements [5,6] or spatio-temporal regularization constraints have also been taken into account [7,8].

In this paper, we propose a variational formulation to accurately estimate the motion vector field in a image sequence introducing a second order Taylor expansion of the flow in the energy function to be minimized. This feature allows us to simultaneously obtain, in addition, an estimation of the partial derivatives of the motion vector field. This idea is quite interesting in the field of fluid dynamics, since partial derivatives is an important source of information in this field as they appear in the Navier-Stokes equations and other intersting parameter such as divergence, vorticity, strain rate tensor and dissipation rate. Moreover, since a regularity constraint is also imposed on the second order flow parameters, the estimated flow will better preserve the continuity behavior assumed in fluid dynamics.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 646–653, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The paper is organized as follows. In section 2, we describe the details of our variational approach and the way we have adapted the energy function to be minimized in order to directly estimate significative fluid parameters such as vorticity and strain rate tensor components. In section 3, we present numerical experiments on synthetic and real data and we analyze the behavior of the proposed method in comparison with other standard techniques described in the literature. Finally, in section 4, we present the main conclusions of the paper.

2 A Variational Approach for Second Order Motion Estimation

To estimate the optic flow of a given sequence we propose a variational approach based on the minimization of an energy function $E(\tilde{\mathbf{u}})$ defined for each point $\mathbf{x_0}$ with the special fact that it depends not only on the displacement vector components $\mathbf{u} = (u, v)^T$, but also on their partial derivatives $\tilde{\mathbf{u}} = (u, v, u_x, u_y, v_x, v_y)^T$, as shown in the following equation:

$$E(\tilde{\mathbf{u}}) = E(u, v, u_x, u_y, v_x, v_y) =$$
$$\int_{\Omega(\mathbf{x_0})} K_\sigma(\mathbf{x} - \mathbf{x_0}) \left(I_1(\mathbf{x}) - I_2(\mathbf{x} + (u, v)^T + \begin{pmatrix} u_x & u_y \\ v_x & v_y \end{pmatrix} (\mathbf{x} - \mathbf{x_0})) \right)^2 d\mathbf{x} \quad (1)$$

where $I_1(\mathbf{x})$ is the first image of the sequence and $I_2(\mathbf{x})$ is the following frame, where the time increment from frame to frame is assumed to be normalized ($\Delta t = 1$), $K_\sigma(\mathbf{x} - \mathbf{x_0})$ is a Gaussian kernel with standard deviation σ which weighs the pixels in the domain $\Omega(\mathbf{x_0})$ centered at point $\mathbf{x_0}$. Hence, the goal is to find the components of vector $\tilde{\mathbf{u}}$, that is, the displacement and its partial derivatives such that minimize the error between $I_1(\mathbf{x})$ and $I_2(\mathbf{x})$ displaced by the unknown flow.

At a first glance, the dependence on the partial derivatives (u_x, u_y, v_x, v_y) might seem a redundant operation since they can be computed from the obtained motion vector field. However, it can be shown that numerical differentiation of the estimated motion vector field yields to inaccurate results mainly due to undesired numerical error amplification [9].

2.1 Energy Minimization

In order to be able to obtain the six components vector of unknowns $\tilde{\mathbf{u}} = (u, v, u_x, u_y, v_x, v_y)^T$ that minimize Eq. 1 we formulate the solution at step $n+1$ as a function of the solution at step n and the six components vector of residuals $\tilde{\mathbf{h}} = (h_u, h_v, h_{u_x}, h_{u_y}, h_{v_x}, h_{v_y})^T$ computed at each step, as shows the following expression:

$$\tilde{\mathbf{u}}^{\mathbf{n+1}} = \tilde{\mathbf{u}}^{\mathbf{n}} + \tilde{\mathbf{h}}\,. \quad (2)$$

Then, introducing Eq. 2 in Eq. 1 we are able to obtain an iterative operator towards the local minimum of the energy function similar to a gradient descent

algorithm [10]. But first, the following approximations are still necessary to simplify the term that depends on $I_2(\mathbf{x})$ in order to obtain our iterative operator:

$$I_2(\mathbf{x}+(u^{n+1},v^{n+1})^T+\begin{pmatrix} u_x^{n+1} & u_y^{n+1} \\ v_x^{n+1} & v_y^{n+1}\end{pmatrix}(\mathbf{x}-\mathbf{x_0}))=$$
$$I_2(\mathbf{x}+(u^n,v^n)^T+\begin{pmatrix} u_x^{n} & u_y^{n} \\ v_x^{n} & v_y^{n}\end{pmatrix}(\mathbf{x}-\mathbf{x_0})+(h_u,h_v)^T+\begin{pmatrix} h_{u_x} & h_{u_y} \\ h_{v_x} & h_{v_y}\end{pmatrix}(\mathbf{x}-\mathbf{x_0}))\,.$$

To simplify notation, let us define I_2^n as:

$$I_2^n(\mathbf{x})=I_2(\mathbf{x}+(u^n,v^n)^T+\begin{pmatrix} u_x^{n} & u_y^{n} \\ v_x^{n} & v_y^{n}\end{pmatrix}\begin{pmatrix} x-x_0 \\ y-y_0\end{pmatrix}) \tag{3}$$

where the vector $\mathbf{x}-\mathbf{x_0}$ has been decomposed into their components $(x-x_0, y-y_0)^T$. The partial derivatives are denoted as:

$$I_{2,x}^n(\mathbf{x})=\frac{\partial I_2}{\partial x}(\mathbf{x}+(u^n,v^n)^T+\begin{pmatrix} u_x^{n} & u_y^{n} \\ v_x^{n} & v_y^{n}\end{pmatrix}\begin{pmatrix} x-x_0 \\ y-y_0\end{pmatrix})\,. \tag{4}$$

$$I_{2,y}^n(\mathbf{x})=\frac{\partial I_2}{\partial y}(\mathbf{x}+(u^n,v^n)^T+\begin{pmatrix} u_x^{n} & u_y^{n} \\ v_x^{n} & v_y^{n}\end{pmatrix}\begin{pmatrix} x-x_0 \\ y-y_0\end{pmatrix})\,. \tag{5}$$

Then, using the new notation, the solution at step $n+1$ can be approximated by its Taylor expansion over $\tilde{\mathbf{u}}^{\mathbf{n}}$ to provide:

$$I_2^{n+1}(\mathbf{x})\simeq I_2^n(\mathbf{x})+\left((h_u,h_v)^T+\begin{pmatrix} h_{u_x} & h_{u_y} \\ h_{v_x} & h_{v_y}\end{pmatrix}\begin{pmatrix} x-x_0 \\ y-y_0\end{pmatrix}\right)^T\begin{pmatrix} I_{2,x}^n(\mathbf{x}) \\ I_{2,y}^n(\mathbf{x})\end{pmatrix}\,. \tag{6}$$

Finally, by using vectorial notation $I_2(\mathbf{x})^{n+1}$ can be expressed as:

$$I_2^{n+1}(\mathbf{x})\simeq I_2^n(\mathbf{x})+\hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x})^T\cdot\tilde{\mathbf{h}} \tag{7}$$

where

$$\hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x})=(I_{2,x}^n(\mathbf{x}),I_{2,y}^n(\mathbf{x}),(x-x_0)I_{2,x}^n(\mathbf{x}),(y-y_0)I_{2,x}^n(\mathbf{x}),$$
$$(x-x_0)I_{2,y}^n(\mathbf{x}),(y-y_0)I_{2,y}^n(\mathbf{x}))^T\,. \tag{8}$$

Hence, introducing Eq. 7 in Eq. 1 we obtain the approximation for the energy function shown in Eq. 9, where, in addition, we introduce a regularization term in the energy function weighted by the parameter α. The role of this regularization term is to provide at every point a smooth vector field, by means of an additional constraint on the norm of the vector $\tilde{\mathbf{h}}$, which is forced to be as small as possible. In this sense, the parameter α determines the importance of this additional constraint on the vector field.

$$\tilde{E}(\tilde{\mathbf{h}})=\int_{\Omega(\mathbf{x_0})}K_\sigma(\mathbf{x}-\mathbf{x_0})\left(I_1(\mathbf{x})-I_2^n(\mathbf{x})-\hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x})^T\cdot\tilde{\mathbf{h}}\right)^2 d\mathbf{x}+\alpha\int_{\Omega(\mathbf{x_0})}\parallel\tilde{\mathbf{h}}\parallel^2 d\mathbf{x}. \tag{9}$$

This formulation allows us to easily obtain an analytical expression to compute the local minimum of the energy as a function of $\tilde{\mathbf{h}}$:

$$\nabla \tilde{E}(\tilde{\mathbf{h}}) = -2 \int_{\Omega(\mathbf{x_0})} K_\sigma(\mathbf{x}-\mathbf{x_0}) \left(I_1(\mathbf{x}) - I_2^n(\mathbf{x}) - \hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x})^T \cdot \tilde{\mathbf{h}} \right) \hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x}) d\mathbf{x} + 2\alpha \int_{\Omega(\mathbf{x_0})} \tilde{\mathbf{h}} d\mathbf{x} = \mathbf{0} \tag{10}$$

which is equivalent to:

$$\int_{\Omega(\mathbf{x_0})} K_\sigma(\mathbf{x}-\mathbf{x_0}) \left(I_1(\mathbf{x}) - I_2^n(\mathbf{x}) \right) \hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x}) d\mathbf{x} = \int_{\Omega(\mathbf{x_0})} \left(K_\sigma(\mathbf{x}-\mathbf{x_0}) (\hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x}) \cdot \hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x})^T) + \alpha \mathbf{I} \right) \tilde{\mathbf{h}} d\mathbf{x} \,. \tag{11}$$

This system of equations can be expressed with the standard matrix notation $\mathbf{A}\mathbf{x} = \mathbf{b}$, where the vector of unknowns in this case is the vector $\tilde{\mathbf{h}}$, while the system matrix and the independent term can be computed as:

$$\mathbf{A} = \int_{\Omega(\mathbf{x_0})} \left(K_\sigma(\mathbf{x}-\mathbf{x_0}) (\hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x}) \cdot \hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x})^T) + \alpha \mathbf{I} \right) d\mathbf{x} \,. \tag{12}$$

$$\mathbf{b} = \int_{\Omega(\mathbf{x_0})} K_\sigma(\mathbf{x}-\mathbf{x_0}) \left(I_1(\mathbf{x}) - I_2^n(\mathbf{x}) \right) \hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}(\mathbf{x}) d\mathbf{x} \,. \tag{13}$$

The solution of the system of equations is given by $\tilde{\mathbf{h}} = \mathbf{A}^{-1}\mathbf{b}$, so we only have to invert the 6×6 matrix $\mathbf{A}$ or use any other algorithm to solve the system of equations. We observe that if $\alpha > 0$, matrix $\mathbf{A}$ is positive definite and therefore the system of equations is well-posed (it has a unique solution). It means that the parameter α avoids instabilities in the solution of the linear system of equations.

2.2 Numerical Implementation

Next, we describe the main steps we followed to derive an efficient algorithm of the method proposed in section 2, including the numerical considerations involved on the computation of integrals and spatial derivatives in the proposed method.

As it can be seen in Algorithm 1, our variational approach starts from an initial estimation of the motion vector field $\mathbf{u}^0$ which can be obtained using any other optic flow estimation method. Then, we start the iterative procedure towards the local minimum of the energy function given in Eq. 1. At each iteration we check the convergence of our algorithm in order to discard the solutions that do not provide the optimal response.

This algorithm is then applied for every point in the image (or a grid at a given scale) to obtain the desired first and second order flow parameters estimation, using information within a neighborhood of that point determined by the σ, parameter of the Gaussian kernel. The initialization of the energy at the first step requires the computation of the image derivatives. In our implementation, we have used central finite differences because it is a good compromise between an easy implementation, low time of computation and low error propagation [9].

Algorithm 1. Implementation of the second order flow estimation method

Initialization of the parameters: σ (window size), α, $\tilde{\mathbf{u}}^0 = \mathbf{0}$, $E(\tilde{\mathbf{u}}^0)$, N_iter, $Failures = 0$, $MaxFailures$
Initial Estimation of the motion field $\mathbf{u}^0 = (u, v)$ with any estimation flow technique

for $n = 0$ to N_iter **do**
  Update $I_2^n(\mathbf{x})$ and $\hat{\mathbf{I}}_{\mathbf{2}}^{\mathbf{n}}$ using $\tilde{\mathbf{u}}^n$
  Computation of the energy $E(\tilde{\mathbf{u}}^n)$, $\mathbf{A}$ and $\mathbf{b}$
  Solve system of equations $\tilde{\mathbf{h}} = \mathbf{A}^{-1}\mathbf{b}$
  if $E(\tilde{\mathbf{u}}^n) \leq E(\tilde{\mathbf{u}}^{n-1})$ **then**
    $\tilde{\mathbf{u}}_{optimized} = \tilde{\mathbf{u}}^n + \tilde{\mathbf{h}}$ {The method converges towards the solution}
    Update $\tilde{\mathbf{u}}^{n+1}$
    $Failures = 0$
  else if $E(\tilde{\mathbf{u}}^n) \geq E(\tilde{\mathbf{u}}^{n-1})$ and $Failures \leq MaxFailures$ **then**
    Update $\tilde{\mathbf{u}}^{n+1}$
    Reject this solution as $\tilde{\mathbf{u}}_{optimized}$ {The method diverges from the solution}
    $Failures++$
  else
    Exit
  end if
end for

3 Results

In order to show the performance of the proposed approach, we present here the numerical experiments we have performed to evaluate the accuracy of the method, working on both synthetic and real data. It is interesting to remark that the initial estimation of the flow can be obtained by any desired method, which must be, at least, a rough approximation of the underlying motion vector field so that the algorithm is able to converge. In our work, we use the method described in [11] to achieve such initialization.

3.1 Experiment 1: Yosemite Sequence

In our first experiment we use the well-known Yosemite sequence to evaluate the performance of our approach. From a qualitative point of view, it is interesting to notice that the vector field obtained with our approach is usually smoother than the initial estimation. This is more clearly seen on Fig. 1, where on the left side we represent the angular error between the groundtruth and the initial estimation, while on the right we show the angular error for our response. As it can be seen, this error feature is lower and more regular with our method. From a quantitative point of view, table 1 shows some statistics on the euclidean error (Eq. 14) and the angular error (Eq. 15) for that sequence according to the ideas in [12]. The error was computed removing the clouds in order to be able to provide a meaningful error measure, following the recommendation in [13]. As it

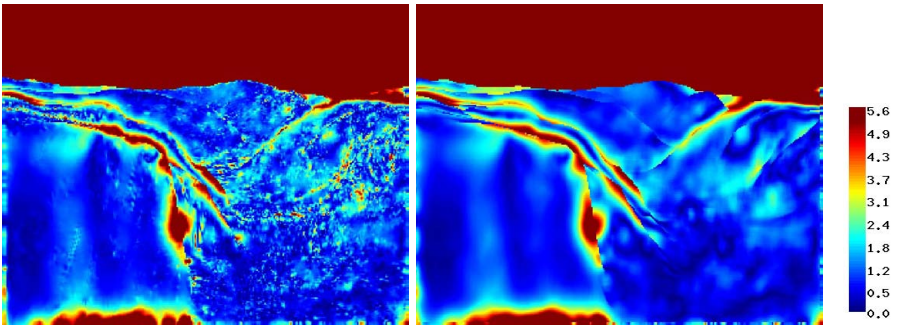


Fig. 1. On the left, angular error between the groundtruth and the initial estimation of the vector field. On the right, the vector field obtained with our method.

Table 1. Quantitative error measures obtained for the initial estimation and for our method

	Angular Error	Std. Dev.	Euclidean Error	Std. Dev.
Initialization	1.6696	1.8765	1.6598	0.01
Our Method	1.5432	1.8467	1.6598	0.01

can be seen in the referred table, the error rate is lower once we use our method to process the initial vector field.

$$RMSVD = \frac{1}{N}\sum_{i=1}^{N} |\mathbf{u}_i - \mathbf{u}_{ref\ i}| \,. \tag{14}$$

$$\psi E = \frac{180}{N\pi}\sum_{i=1}^{N} \arccos(\mathbf{u}_i \cdot \mathbf{u}_{ref\ i}) \,. \tag{15}$$

3.2 Experiment 2: Satellite Sequence

Finally, we present the results we obtained with our approach using real satellite image sequence provided by the Laboratoire de Meteorologie Dynamique (LMD) from Paris (France). This data tracks the hurricane Vince that became the first known tropical cyclone to reach the Iberian Peninsula between October 8th and 11th, 2005. The main interest in processing this kind of data is to estimate the cloud motion between consecutive frames.

In Fig. 2 we compare the vector field obtained with our approach (represented in yellow, light arrows) with that used to initialize our method (represented in red, dark arrows). As it can be seen, the vector field obtained is more regular than the initial estimated flow.

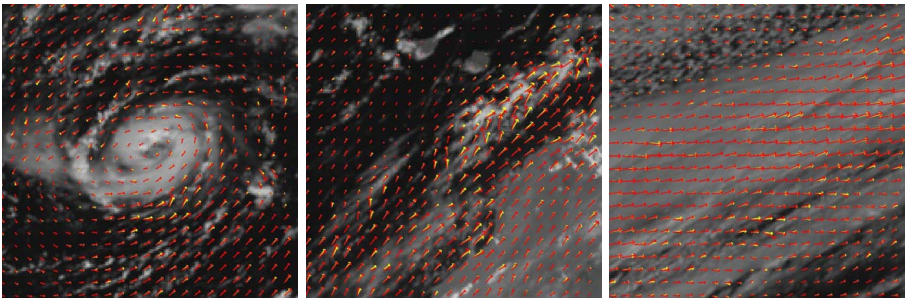


Fig. 2. Details of the motion vector field in the satellite sequence. In red (dark arrows), we represent the method used to initialize our algorithm and in yellow (light arrows) we display the vector field obtained with our method.

4 Conclusions

In this paper, we have presented a new variational method to estimate the motion vector field in a image sequence. In this sense, our variational method improves the quality of the motion vector field used as initial estimation introducing a regularity constraint on the first and second order moments of the motion vector field to be estimated. Our quantitative results on synthetic data show the encouraging results we obtain using our method as a post-processing step for the regularization process. The performance of our method is also tested with real satellite image sequences, where the vector fields we obtain also present a regular behavior.

Acknowledgements

This work has been funded by the European Commission under the Specific Targeted Research Project FLUID (contract no. FP6-513663).

References

1. Horn, B., Schunck, B.: Determining optical flow. MIT Aritificial Intelligence Laboratory (1980)
2. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. Seventh International Joint Conference on Artificial Intelligence, Vancouver (1981) 674 – 679
3. Nagel, H.H., Enkelmann, W.: An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. IEEE Trans. Pat. Anal. and Mach. Intell. **8** (1986) 565–593
4. Aubert, G., Deriche, D., Kornprobst, P.: Computing optical flow via variational techniques. SIAM Journal on Applied Mathematics **60**(1) (1999) 152–182
5. Mémin, E., Pérez, P.: Dense estimation and object-based segmentation of the optical flow with robust techniques. IEEE Transactions on Image Processing **7**(5) (1998)

6. Alvarez, L., Weickert, J., Sánchez, J.: Reliable estimation of dense optical flow fields with large displacements. International Journal of Computer Vision **39**(1) (2000) 41–56
7. Weickert, J., Schnorr, C.: A theoretical framework for convex regularizers in pde-based computation of image motion. Draft, Departament of Mathematics and Computer Science (2000)
8. Weickert, J., Papenberg, N., Bruhn, A., Brox, T.: High accuracy optical flow estimation based on a theory for warping. Volume 4. (2004) 25–36
9. Raffel, M., Willert, C., Kompenhans, J.: Particle Image Velocimetry. A Practical Guide. Springer Verlag (1998)
10. Fletcher, R., Reeves, C.: Function minimization by conjugate gradients. Computer Journal **7** (1964) 149–154
11. Alvarez, L., Castaño, C., García, M., Krissian, K., Mazorra, L., Salgado, A., Sánchez, J.: Symmetric optical flow. In: Proceeding of EuroCast, Las Palmas, Spain (2007)
12. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. IJCV **12**(1) (1994) 43–77
13. Black, M.: http://www.cs.brown.edu/people/black/sequences/yosfaq.html (2006)

An Application of Optical Flow: Slow Motion Effect on Streaming Image Sequences

Roman Dudek, Carmelo Cuenca, and Francisca Quintana

Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria, Spain

Abstract. In this paper we describe an application of the optical flow in order to create a slow motion effect and frame rate conversion on streaming image sequences. Our results show that using the optical flow based method the image sequence shows much less artifacts than than in traditional interpolation or mixing methods.

1 Introduction

In traditional film shooting, slow motion effect is achieved by shooting the film footage at frame rate higher than the assumed projection frame rate, this being normally 24 frames per second (fps). To achieve a slow motion shot showing a real movement at, for example $\frac{1}{3}$ of the real time speed, the camera needs to be run at $3 \times 24 = 72$ fps. This approach has obvious limitations: The possible frame rate of a camera has mechanical limits, to begin with. In the case of video image sequences, normally it is impossible to choose filming frame rate at all.

A different, obviously desirable approach would be shooting at normal rate, and then increasing (or arbitrarily re-mapping) the number of frames of the sequence so the movement would appear at speed different to that of the shooting.

Another important reason for such re-mapping would be video/film standards conversion: Multiple frame rates are used in different countries and formats: 24 fps for film, 25 fps for PAL video (Europe), 29.97 fps for NTSC (USA, Japan), 60 fps for HD 720 line standard. Frequently, media produced in one format needs to be converted into different format, for international sports broadcasts, for example.

2 Known Approaches for Slow Motion Simulation

The simplest approach consists of repeating each frame multiple times. However, the smoothness of the movement suffers considerably. With high slowdown factor, result resembles a slideshow of sequence of still images.

Certain improvement over repeating frames is a simple mixing of pairs of consecutive images: Image 0 (I^0) cross fades to image 1 (I^1), then the I^1 cross fades into image 2 (I^2) and so on. However, important artifacts arise in the slowdown of any rate: Double edges on moving objects are visible, and their formation tends to disappear and reappear periodically.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 654–659, 2007.
© Springer-Verlag Berlin Heidelberg 2007

To create a real impression of a smooth movement in the slowed down scene, the recreated intermediate images need to interpolate smoothly the movement the scene actually contains. If an object moves from one position in an I^0 to another position in an I^1, the intermediate images need to interpolate intermediate positions of the object. However, an object can change its form between frames, not only its position. Apart of non-trivial inter-frame changes like progressing occlusion of an object, more subtle changes, like shadow sliding over an object, can happen.

Creating intermediate frames between a pair of not necessarily exactly matching states of the scene is a problem known as early as with the first animated movies, where a leading artist created "key frames" of a character, while auxiliary artists worked on so called "tweening" short for "in betweening": Manual creation of all the intermediate frames to achieve a smooth animation.

More recently, with assistance of computers, so called "morphing" is used to transform certain object into a different one, like transforming a turtle into a bird in a fantasy movie. An artist determines manually which feature an I^0 will be mapped onto which feature of an I^1. To create the intermediate animation, the computer then distorts gradually the I^0 till its features match the positions in the I^1. To get a smoother transition, also the I^1 is distorted to first to map its features to their positions in the I^0, and then its distortion is gradually reduced.

Then, both sequences are gradually mixed (cross faded), so the features that could not be exactly matched will at least gradually fade out (or fade in).

Obviously, the morphing process could be used for creating the intermediate frames of a slowed down sequence. However, manual mapping of features would be extremely laborious, so need for automated feature mapping arises.

Optical flow based methods offer themselves to map the moving features of an image sequence, and drive the morphing process to create the new intermediate frames.

Multiple optical flow calculation methods, and their modifications, were devised over many years of investigation. To choose a method for our purpose, the method should full these criteria:

- Large displacements need to be resolved correctly. At typical 2048×1556 digitalized film resolution, a movement that crosses the frame in 5 seconds (commonly accepted limit for camera pan speed), leads to a displacement of 17 pixels per frame.
- It should be fast. Processing times of more than one second per frame of video are usually not considered acceptable in video postproduction.
- It should provide dense vector field, ideally at every point of image.
- It should be predictable and stable. It should not lead to dramatically incorrect results even in a small zone of the scene. If some movement is misinterpreted, it is better if this error persists over the sequence than an error randomly popping in and out when crossing some threshold of the method. For this reason, any hard thresholds in the method should be avoided. We refrained from use of pattern matching based methods because of this reason, as they usually imply making hard decisions between different, apparently similarly good solutions.

3 The Optical Flow Calculation Method

Based on comparison studies by [1,2,3], and on our trials with a few methods, we have chosen a variation of the most classical of all algorithm of all: The Horn and Shunck [4], extended by a multi-pyramidal scheme to account for large movements. We also added a modification into the calculation of the luminance gradient over time, improving considerably the method's results. On a synthetic test sequence with known correct solution (the Yosemite sequence), the result average error was better than the best algorithm evaluated in [2] in their year 1994 survey, with practically no added complexity. (Better methods were meanwhile devised, however at considerable complexity cost). More importantly, this method provides visually the best results of methods we tested on real world image sequences - what was our actual goal in this work.

In our multi-pyramidal scheme, the I^0 and I^1 image pair is repeatedly filtered and reduced to half resolution in each dimension, till the largest expected movement becomes of approximately 1 pixel in size. Four steps would, for example accommodate for 16 pixels of maximum movement. Then, beginning with the deepest (most reduced resolution) level, the modified Horn and Shunck algorithm is iterated. Performed the iterations, the obtained $(u, v)^T$ field is doubled in size using bicubic interpolation. Also, the magnitude of the $(u, v)^T$ vectors is doubled to accommodate for the change of scale. This doubled $(u, v)^T$ is then used as initial value set for iterating at the next level. The process of iterating-doubling is repeated until we reach the full resolution image.

Our modification to the Horn and Shunck algorithm consists in a change of the way the temporary derivative It is calculated. The original method calculates it as

$$I_t(x, y) = I^1(x, y) - I^0(x, y). \tag{1}$$

Then, the term to minimize is calculated

$$I_x^0\overline{u} + I_y^0\overline{v} + I_t(x, y) = I_x^0\overline{u} + I_y^0\overline{v} + I^1(x, y) - I^0(x, y). \tag{2}$$

In our modification, instead of using linear extrapolation to estimate the luminance change, we actually do a lookup of it offsetting the point at I^1 at which the luminance is compared, and replacing the original terms $(u, v)^T$ by zero to compensate this offsetting

$$I^1(x + \overline{u}, y + \overline{v}) - I^0(x, y). \tag{3}$$

Intuitively, this represents looking up where the point of I^0 is supposed to move when in the image I^1 , and compare the luminance with the expected one. Based on this comparison, and the I^0 image slopes, the vector is then corrected to get nearer to a luminance match.

This modification has important advantages over the original method: First, it is not anymore limited to vectors of sub-pixel size, as the I^1 lookup can be done at arbitrary distance, provided that the previous values of $(\overline{u}, \overline{v})^T$ are themselves precise to less than one pixel error. In this aspect, the method resembles a pattern

matching, and simplifies considerably the implementation of the multi-pyramidal scheme. Second important advantage is that this formulation facilitates the diffusion along edges, providing a solution to the aperture problem: Once the right offset for an edge displacement is found, the term will remain zero if point is moved along the edge. However, the term will rise (or drop) sharply when the point is moved perpendicularly to the edge, causing the vector to move back to a stable state. In consequence, the vector is allowed to drift freely along the edge, while opposing strong resistance to any movement perpendicular to the edge. This seems to amount to an anisotropic diffusion mechanism.

4 The Morphing Based Image Interpolation

Once the optical flow from I^0 to I^1 is calculated, we need to perform the morphing process.

Let the $\lambda : [0, 1]$ be the relative time of the intermediate image I^λ; between the I^0 and I^1. We need to warp (distort) both images, in proportion of λ, or $(1-\lambda)$ respectively, to align the "forward warp" of I^0 and the "backward warp" of I^1 in a certain intermediate position.

The optical flow field basically determines where each point of the I^0 moved when reaching the time corresponding to the frame I^1.

Not so obviously, it is actually much simpler to warp the I^1 into I^0 using the I^0 to I^1 optical flow.

By definition of the optical flow, $I^0 = I^1(x + u(x, y), y + v(x, y))$. In consequence, for full warping (at $\lambda = 0$), for each resulting pixel, we need to simply lookup the value of the I^1 in the position $I^1(x + u, y + v)$.

To achieve partial warping for $\lambda > 0$, we scale down the vector $(u, v)^T$: $I^1(x + (1-\lambda)u, y + (1-\lambda)v)$. For λ rising from 0 to 1, we will obtain first the I^1 warped to match the I^0, then gradually "unwarping" to its final (and original) form for $\lambda = 1$.

For the I^0, straightforward solution would be moving each pixel of the I^1 image gradually in the direction of its corresponding movement vector. However, this is not easy to implement, especially with sub-pixel precision, as the destination positions are normally non-integer, falling in-between of pixels of the result. One way to perform this "moving of pixels" would be drawing a fine polygon mesh, with a vertex at each pixel. However the polygon count would be extremely large: about 6000000 triangles for the above mentioned Cine 2K resolution. Possibly, sparser mesh could be used, as the field is typically smooth. We plan to test this option in the future.

Another way to solve this problem is searching for each pixel its neighborhood to find which pixel (or better, an interpolated sub-pixel position) would map its vector into our current position. While this search could have multiple or no solutions, normally in case of relatively smooth vector fields this solution can be found numerically very fast, using the $I^0(x - u, y - v)$ as the first estimation for the point that should map into our position $(x, y)^T$. Then, we calculate where the candidate point actually maps, and use the resulting error as correction

for improving the candidate position. We found that in practice this method converges very fast, (from 1 to 5 steps for maximum position error of 0.01 pixel size) and fails only rarely, on some extreme movements in the scene. In these cases, we do not mix the images and use only the other from the couple of warps we are creating. Finally, a simple weighted mix is performed between the two warped results:

$$I^{\lambda}(x,y) = (1-\lambda)I^{0}(x',y') + \lambda I^{1}(x+(1-\lambda)u(x,y), y+(1-\lambda)v(x,y))$$
$$\text{where} (x',y')^{T} \text{is such as}$$
$$(x,y)^{T} = (x'+\lambda u(x',y'), y'+\lambda v(x',y'))^{T}. \quad (4)$$

It is worth mentioning that if the factor of speed change is not an integer number (being 29.97/25 for NTSC to PAL conversion, for example), most of the frames of the original sequence would typically not appear at all in the resulting sequence, being actually all frames the interpolated ones.

5 Results and Further Work

We have obtained visually convincing results for variety of real word scenes, being the results vastly superior to straightforward frame repeating or mixing. Only few artifacts can be found in our results, especially in zones where large movements in conflicting directions overlap. The quality of the optical field seems to directly define the quality of the results, so most effort should be spent on improving the optical flow algorithm. Unfortunately, nearly perfect solution would possibly require the algorithm to be capable to make very advanced suppositions about the images, including these of type "how many fingers is supposed to have a person on its hand".

Figure 1 shows four images of a video sequence. The two upper images are an example of two consecutive images which we want to use for interpolation in order to obtain a slower sequence. The down-left image is an image resulted of the application of the mixing method. It is possible to observe blurring artifact around the edges of the objects. The down-right image is an image obtained using proposed optical flow and morphing based interpolation. In this case, edges of the object are not mixed giving a better visual perception. Several additional examples could be got in [5].

The computation time, with highest sensible iteration counts, is about 400ms per I^0 and I^1 pair of full PAL resolution (720×576 pixels), in a CPU implementation. With lover count of iterations, causing only minor reduction in quality, 100ms is sufficient, on a 2×Pentium IV, 2.8Ghz computer. Once obtained the optical flow field, the rendering of each intermediate frame costs about 60ms.

However, all calculations used in our algorithm that have relevant computational cost, seem to be well suited for implementation in graphical hardware (GPU). Judging by experience on similar problem implementations, about 10× speedup could be expected by using the GPU, bringing the algorithm definitively into real-time domain.



Fig. 1. Two consecutive images and a sample result of the slow motion computation

Explicit solution to occlusions problem could be also developed. Again, the success of this method would be extremely sensitive to the optical flow field quality, and both directions of optical flow would need to be calculated, to cover both cases where an object is occluded and revealed.

References

1. Álvarez, L., Weickert, J., Sánchez, J.: Reliable Estimation of Dense Optical Flow Fields with Large Displacements. International Journal of Computer Vision 39(1), 41–56 (2000)
2. Beauchemin, S.S., Barron, J.L.: The Computation of Optical Flow. ACM Computing Surveys 27(3), 433–467 (1995)
3. McCane, B., Novins, K., Crannitch, D., Galvin, B.: On Benchmarking Optical Flow. Computer Vision and Image Understanding 84(1), 126–143 (2001)
4. Determining Optical Flow, AI Memo 572, Massachusetts Institue of Technology (1980)
5. `http://www2.dis.ulpgc.es/~ccuenca/eurocast2007` (2007)

Comparing Self-calibration Methods for Static Cameras*

J. Isern González, J. Cabrera Gámez, J.D. Hernández Sosa,
and A.C. Domínguez Brito

Instituto Universitario de Sistemas Inteligentes y Aplicaciones Numéricas en Ingeniería (IUSIANI)
Universidad de Las Palmas de Gran Canaria
Spain, LAS PALMAS 35017
{jisern,jcabrera,dhernandez,adominguez}@iusiani.ulpgc.es

Abstract. Many methods have been developed in the last few years to self-calibrate cameras, but few works have addressed the comparison of such methods to provide the user with hints on the suitability of certain algorithms under particular circumstances. This work presents a comparative analysis of four self-calibration methods for cameras which only rotate. This paper concentrates on the stability, the accuracy in the estimation of each parameter and the computational cost. This study has been carried out with real and simulated images. The experiments have shown that the optic center is the most unstable parameter for all methods and that the greatest discrepancies among the estimated values appear with the scale factors. Also, there are no correspondence among image disparity and parameters error. Finally, the results returned by any of these methods are comparable in terms of accuracy with those provided by a well-known manual calibration method.

1 Introduction

Camera calibration is a procedure that tries to know how a camera projects a 3D object on the screen. This process is necessary in those applications where metric information of the environment must be derived from images.

Classical calibration methods use a known pattern as a reference to manually calibrate a camera. Meanwhile these algorithms can achieve accurate results, they do not allow on-line re-calibration if, for example, the internal configuration of the camera is changed due to a new visual task. Self-calibration methods have been proposed to overcome these problems permitting an automatic calibration. These methods use the correspondence of the same static points acquired from different locations.

In the literature on camera calibration there is a lack of works comparing different kinds of methods. The very few works that analyze self-calibration

* This research has been partially funded by the Canary Islands Autonomous Government (proj. PI2003/160) and the Univ. of Las Palmas de G.C. (proj. UNI2005/18).

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 660–667, 2007.
© Springer-Verlag Berlin Heidelberg 2007

methods have concentrated on the study of the influence of some factors [1] [2] [3] (misalignment of the focal point and axis of rotation, lens distortion) or the identification of critical movements [4] [5] [3], but there are not any studies where the accuracy and stability of these methods is analyzed.

The present study is an extension of [6] [7] in some specific aspects. In these works a set of classical calibration methods were studied using real and simulated data. The accuracy, the stability and the influence of specific factors on the calibration were analyzed.

In the present work a similar study has been carried out. In this case, a set of four methods of self-calibration for cameras which only rotate (without any translation movement) has been analyzed. These methods have been compared and their stability, accuracy in the estimation of the camera internal configuration and computational cost have been studied. The experiments have been carried out with a real camera and with a simulator.

This paper is organized as follows. The next section will introduce the camera model and the methods that have been used in the comparative study. Section 3 will present some of the experiments and comment on their results. The last section is devoted to a summary of the main conclusions of this work.

2 Self-calibration Methods

Self-calibration methods can be classified according to the type of camera movement. In the first case, there are methods that need views acquired from different places and with different orientations [8] [9]; others need the realization of specific displacements (combining pure translations with pure rotations) [10] or are prepared for displacements on one plane [11]. Finally there are methods that only permit rotations [12] [13] [14] [15].

This work is concentrated on four methods of the last group (without translation movement). All these methods follow the pin-hole camera model. The parameters obtained from the calibration process can be classified into two types: *intrinsics*, in other words, those that define the camera configuration (internal geometry and optic features); and *extrinsics*, in other words, the relation (rotation and translation) between the views of the scene.

Self-calibration methods usually formulate the transformation from 3D coordinates to 2D coordinates using a matrix notation: $\lambda \cdot m = P \cdot M$ where M is 3D coordinates and m image coordinates of the point and λ is a non-zero scale factor. Finally, the 3x4 projection matrix (P) can be split up into two matrices: $P = A \cdot D$, one for extrinsic parameters ($D = [R|T]$, where R is the rotation matrix and T is the translation array) and another for intrinsic parameters:

$$A = \begin{pmatrix} k_x & -k_x \cot \Omega & u_0 \\ 0 & k_y / \sin \Omega & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

where $k_x = d_x f$ and $k_y = d_y f$ are horizontal and vertical scale factors, being d_x and d_y the transformation factors, f the focal length, Ω the skew, and the optical center position is represented by u_0 and v_0.

In the next subsections the characteristics of the methods considered in this study are briefly presented.

2.1 McLauchlan's Method [15]

This method assumes that the camera setup is fixed along a sequence, so only one value is obtained for each intrinsic parameter. A Variable State-Dimension Filter (VSDF) is used to estimate the camera parameters, so that initial values need to be provided. This filter is used to predict the new coordinates of each point and the error in the estimation is used to correct the value of each parameter. The algorithm described by the authors only determines the scale factors and alignment offset angles. The remaining parameters must be known, including the camera displacement.

2.2 Agapito's Methods [13] [12]

Two versions of this method have been studied in this work. In both cases, the variation of optic center and scale factors is allowed along the sequence, i.e. camera can zoom in and out. They present some limitations as the fact that skew, pixel ratio and distortion must be fixed and known when roll movement is not possible. Theoretically, a minimum of three images are necessary to perform the calibration process.

The first version [13] is a **linear algorithm** which is based on the obtention of the absolute conic from the homografies between each image and reference image (the first one). Cholesky factorization is used to extract the camera parameters.

The second version [12] is an **iterative algorithm** that adds an optimization stage using the results of the linear version as initial guess values.

2.3 Seo's Method [14]

This algorithm allows the variation of optic center and scale factors along a sequence of images. The method is based on the obtention of the absolute conic from the homographies between each pair of views. It needs skew and distortion to be known and fixed. First a non-iterative algorithm is used to get a initial guess for scale factors and then, an optimization procedure is used to obtain final values of scale factors and optic center. Theoretically, a minimum of four images are necessary to carry out the calibration process.

3 Experimental Results

The experiments discussed in this paper have focused on characterizing the global accuracy, the stability and accuracy of the estimation of the internal camera parameters and the computational cost of the algorithm. The error in points reconstruction (mean distance between real and estimated image coordinates) has been measured to know their global accuracy. The average distance between the

ground truth value and the estimated value has been used to calculate the error in each parameter. The variability in the estimation of the intrinsic parameters along the sequence has been measured to know the stability of the methods.

The experiments were performed with both a real camera and a simulator. When the calibration is performed with real data, ground truth values of each parameter of the camera model are unknown, so it is impossible to determine the accuracy with which these parameters are obtained. This difficulty was overcome using a simulator which was developed in Matlab. In the simulator, the experiment was performed injecting a gaussian noise in 2D coordinates of the points and the camera lens was modeled with a little distortion. Working with real data, only the stability and global accuracy could be studied for each method.

The camera used in the experiments with real data was a Sony EVI-G21, which has motorized pan, tilt and zoom as well as auto-focus, but this was disabled in the experiments. This camera was connected to an Imaging IC-PCI-COMP frame-grabber card. The image resolution was set to 768x576 pixels.

A planar pattern with 20x15 points, which was placed 1m away from the camera, was used to obtain the reference points of the scene. Although, 3D coordinates of the points were known, this information was not managed by the self-calibration methods to calibrate the real camera in the experiments. The reasons for using a pattern were three: facilitate points correspondence avoiding false matches, their results can be compared with a manual calibration method (Batista's method [16]) and error in the reconstruction of 3D coordinates of points can be measured.

3.1 Stability Analysis

When only the extrinsic parameters (camera orientation) vary in a sequence used to calibrate a camera the estimated values for the intrinsic parameters in each image of this sequence should be the same. The maximum difference of calibration results have been used to estimate the stability.

A sequence of 40 views, where the camera configuration was keeping constant, was used to know the stability of the intrinsic parameters. In the sequence, pan and tilt movements were combined to generate a conic displacement of the view (Fig. 1) to avoid critical movements such as rotation around only one axis.

The experiments carried out showed that, although the camera configuration was kept constant along the sequence, there were variations across all methods

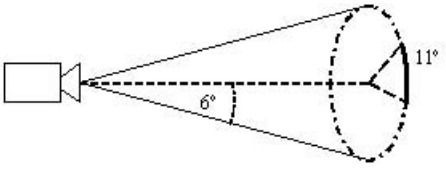


Fig. 1. Camera movement

by intrinsic parameters (Fig. 2). For example, this variability represented 4% of the ground truth value in the optic center when simulated images were used and 3% when real images were analyzed . McLauchlan's method doesn't appear in these graphics because doesn't estimate the optic center. However, the results of the scale factors were very stable. Additionally, the estimated values of intrinsic parameters obtained by the methods had differences among them, especially in the scale factors (4% using simulated data and 2.5% using real data). The two versions of Agapito's method have obtained the same result.

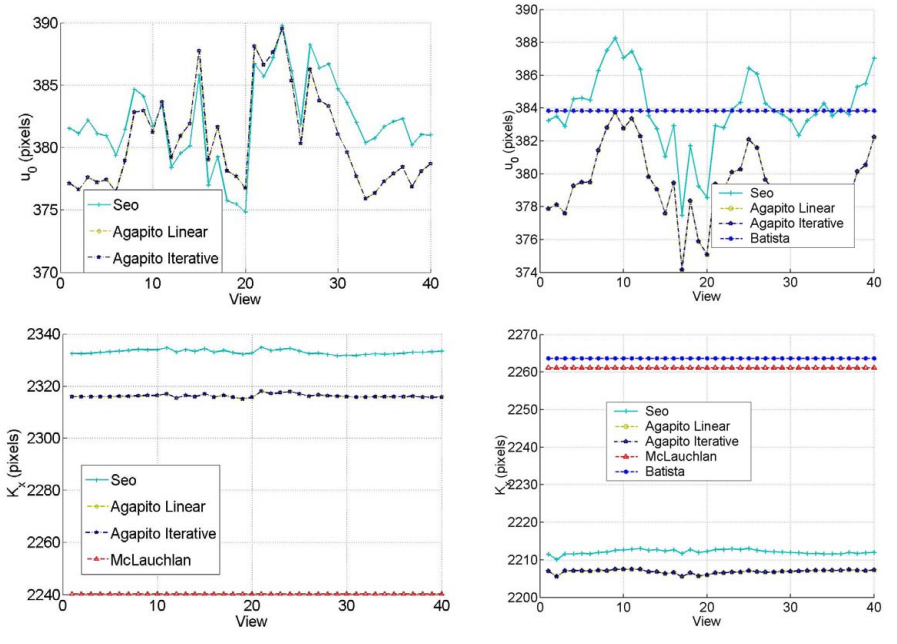


Fig. 2. Optic center in the simulator (top-left) and real camera (top-right). Scale factor in the simulator (bottom-left) and real camera (bottom-right). Ground truth values: $u_0 = 378.3pix.$ and $k_x = 2232pix.$

3.2 Accuracy Analysis

The accuracy of each parameter could be analyzed in the simulator (Fig. 2). It was observed that the versions of Agapito's method obtained values closer to ground truth value of the optic center than Seo's method. McLauchlan's method obtained a high accuracy in scale factor due to the fact that it estimates only one value for all the sequence and the remaining methods studied presented little inaccuracy (4.5%). However, the global error of McLauchlan's method had the highest value and Agapito's method obtained a very low level for this error

Table 1. Error in the reconstruction of 2D coordinates in the simulator

Methods	Agapito (Lin)	Agapito (Iter)	Seo	McLauchlan
Global error (pix.)	0.099	0.099	0.24	0.45

(Table 1). It shows that there is no correspondence between the parameters error and the global error.

In real conditions, the results of these self-calibration methods were compared with a manual calibration method (Batista's method [16])). In general, while the manual method offered more stable results across the sequence, the performance of the self-calibration methods was comparable to that achieved by Batista's method. The self-calibration method that obtained closer values to this method was McLauchlan's method, which turned out to be the most precise method in the simulator.

Finally, the influence of lens distortion and noise on the accuracy was studied individually in the simulator (Fig. 3). Distortion was the factor that had more infuence on the accuracy of the parameters, especially in the methods that allow the variation of intrinsic parameters along the sequence. The influence of the noise wasn't significant in the accuracy of the parameters estimation.

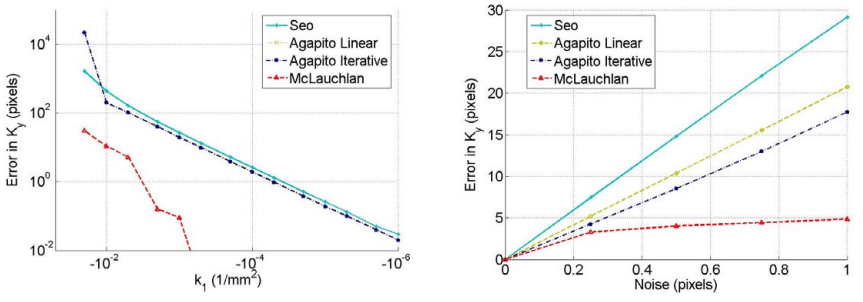


Fig. 3. Influence of lens distortion (left) and noise (right) on the accuracy

3.3 Computational Cost

The effect of the length of the sequence on the accuracy of each parameter was studied. The results (Fig. 4) showed that 20 images were enough to obtain minimum levels of errors in all methods, but McLauchlan's method, although it had less error, needed more images to minimize the error. On the other hand, when the run time was compared, it could be observed that Seo's method had a high computational cost, especially in long sequences.

Also the influnce of the number of points visible in each view was analyzed (Fig. 4). It can be observed that the computational cost of McLauchlan's method has a high dependence on the number of points processed because the size of covariance matrix used in the filter is proportional to the number of points.

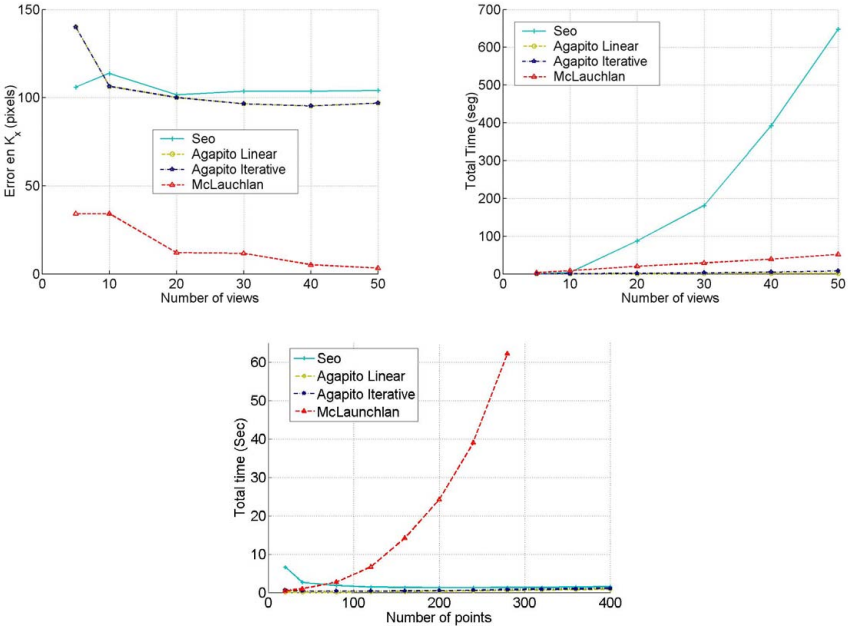


Fig. 4. Influence of the number of views on the scale factor (top-left) and run time (top-right). Influence of the number of points in the run time (bottom).

4 Conclusions

After this study we can conclude that these methods have shown little variations of intrinsic parameters along the calibration sequence when the camera setup was fixed, being the optic center the most sensible parameter. The values estimated by the methods have shown up discrepancies, especially scale factors. McLauchlan's method has turned out to be has the most accurate one, but it is also intractable if the number of points goes up. On the other hand, the computational cost of Seo's method grows exponentially with the sequence length. Also, the results returned by these methods are comparable in terms of stability and accuracy (although they use a simpler camera model) with those provided by a well-known manual calibration method. Finally, there has been no correspondence between the error in the estimation of the parameters and the global error (image disparity) which is the function that tries to minimize all the methods.

References

1. Hayman, E., Murray, D.: The effects of translational misalignment when self-calibration of rotating and zooming cameras. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 25(8), 1015–1020 (2003)

2. Wang, L., Kang, S., Shum, H.Y., Xu, G.: Error analysis of pure rotation-based self-calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(2), 275–280 (2004)
3. de Agapito, L., Hayman, E., Reid, I.: Self-calibration of rotating and zooming cameras. Technical Report OUEL 0225/00, Department of Engineering Science, University of Oxford (2000)
4. Kahl, F., Triggs, B., Åstrom, K.: Critical motions for autocalibration when some intrinsic parameters can vary. J. Math. Imaging and Vision 13(2) (October 2000)
5. Sturm, P.: Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In: Conf. Computer Vision and Pattern Recognition, pp. 1100–1105 (1997)
6. Isern González, J.: Estudio comparativo de métodos de calibración y autocalibración de cámaras. PhD thesis, University of Las Palmas de Gran Canaria, Las Palmas, Spain (July 2003)
7. Isern González, J., Hernández Sosa, J.D., Domínguez Brito, A.C., Naranjo Cabrera, A.: Comparative analysis of calibration methods for a static camera. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, Springer, Heidelberg (2005)
8. Maybank, S., Faugeras, O.: A theory of self-calibration of a moving camera. International Journal of Computer Vision 8(2), 123–151 (1992)
9. Hartley, R.: Estimation of relative camera positions for uncalibrated cameras. In: Sandini, G. (ed.) ECCV 1992. LNCS, vol. 588, pp. 579–587. Springer, Heidelberg (1992)
10. Ma, S.: A self-calibration technique for active vision systems. IEEE Trans on Robotics and Automation 12(1), 114–120 (1996)
11. Faugeras, O., Quan, L., Strum, P.: Self-calibration of a 1d projective camera and its application to the self-calibration of a 2d projective camera. IEEE Transactios on Pattern Analysis and Machine Intelligence 22(10), 1179–1185 (2000)
12. de Agapito, L., Hayman, E., Reid, I.: Self-calibration of a rotating camera with varying intrinsic parameters. In: Proc. British Machine Vision Conference, pp. 105–114 (1998)
13. de Agapito, L., Hartley, R., Hayman, E.: Linear calibration of a rotating and zooming camera. In: CVPR99 (June 1999)
14. Seo, Y., Hong, K.: Auto-calibration of a rotating and zooming camera. In: Proc. of IAPR workshop on Machine Vision Applications, pp. 17–19 (November 1998)
15. Mclauchlan, P., Murray, D.: Active camera calibration for a head-eye platform using the variable state-dimension filter. IEEE Transactions on Pattern Analysis and Machine Intelligence 18(1), 15–22 (1996)
16. Batista, J., Araújo, H., de Almeida, A.: Iterative multi-step explicit camera calibration. In: Proc. Sixth International Conference on Computer Vision ICCV 1998, vol. 15, pp. 709–714 (1998)

Automation of Snakes in Medical Images

Julio Esclarín Monreal and Carolina García Antón

Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria. Spain
jesclarin@dis.ulpgc.es

Abstract. Nowadays time the 3D reconstruction from 2D images has become a tool of daily use for the medical diagnosis. Usually, many tools exist that can make this reconstruction, but when discovering problems in small vessels or areas, as it could be the neck, then tools that detect the borders of the objects of a precise way are needed. For that reasons, in this work, we have used the snake method, that allows to find the contour of small objects with high accuracy. A problem that presents the snake process is that it needs to start from an initial contour, which is usually introduced by hand. This becomes a big problem in medical imaging where the amount of images is very large, and the number of sections of small objects in each image also can be important. In this work we have automated the entrance of this initial contour from an inner point of the object that we want to detect.

Keywords: Snake, 3D Reconstruction, Medical Imaging, XmegaWave.

1 Introduction

In this piece of work we present a method to obtaining the contours of the blood vessels of the neck from stacks of two dimensional angiographic images slices (Volumetric Image) as we can see in the figure 1.

This method works almost automatically from a inside point of one object in the first image that it object appears and it's working until the object disappears.

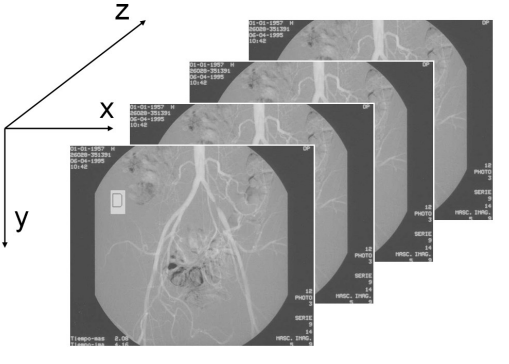


Fig. 1. Stacks of two dimensional angiographic images

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 668–675, 2007.
© Springer-Verlag Berlin Heidelberg 2007

We can think of these slices as serial cross- section a through a scene depicting objects in a digitized 3D world.

2 Tools

2.1 Thin Edges

The classical approximation of the snakes consists of the deformation of an initial contour C_0 that is found around the border of the contour to be detected. The deformation is obtained trying to minimize a functional design where its minimums (local) are obtained of the border of the object.

We use the algorithm ami_snake() that was developed in our laboratory. It is an implementation of the 2D snake technique proposed in [7]. It is a level set method based on the partial differential equation. It is a level set method based on the partial differential equation

$$\frac{\partial u}{\partial t} = g_\sigma(I) div\left(\frac{\nabla u}{\|\nabla u\|}\right)\|\nabla u\| + \lambda \nabla u \nabla g_\sigma \qquad (1)$$
$$u(0,x,y) = u_0(x,y)$$

Where $\lambda \geq 0$, I represents the grey level intensity of the input image, $g\sigma(I)$ is defined by:

$$g_\sigma(I) = \frac{1}{\sqrt{1+\alpha\|\nabla G_\sigma * I\|^2}}$$

with $\alpha > 0$ and $G\sigma * I$ is the convolution of I with the Gaussian kernel

$$G_\sigma(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}},$$

$u_0(x,y)$ represents the characteristic function associate to initial snake contour C_0.

For simplicity we assume that C_0 is a single Jordan curve given by a polygon. So $u_0(x,y)$ is given by

$$u_0(x.y) = \begin{cases} l_1 & if\ (x,y)\ is\ inside\ C_0 \\ l_2 & if\ (x,y)\ is\ outside\ C_0 \end{cases}$$

where l_1 and l_2 are the intensity levels where the snake is included. The snake evolution is provided by equation (1). Indeed if $u(t,x,y)$ is the solution of (1) at time t: Then the associated snake C_t is given by

$$C_t \equiv \partial\left\{(x,y)\ : u(t,x,y) > \frac{l_1+l_2}{2}\right\}$$

that is the boundary of the level set of u associated to the level $(l_1 + l_2)/2$. In particular $c_\infty \equiv \lim_{t \to \infty} C_t$ represents the output contour provided by the equation.

In order to deal with the case where the initial snake C_0 is far away from the target final contour we use a multiscale strategy to help the convergence of the algorithm to the right target contour.

We define $\sigma_n = n\,\sigma_0$ for n = 1; 2,….,N. We note by $u^{\sigma_n}(t, x, y)$ the solution of (1) for $\sigma = \sigma_n$ and the initial datum $u_0^{\sigma_n} = Lim_{t \to \infty} u^{\sigma_{n+1}}(t, x, y)$. $\left(u_0^{\sigma_N} = u_0\right)$. Then, the final contour C_∞ is given by the boundary of the level set $(l_1 + l_2)/2$ of the asymptotic state of $u^{\sigma_1}(t, x, y)$.

2.2 Thin Edges

Numerous edge detectors have been introduced in the last years. Most of them are based on detecting points in the image with a high gradient value. However, in general, they omit the idea that the edge of an object must be a connected line of pixels. Instead of that, these techniques normally use a global threshold value, in order to control the quantity of edge points detected. Due to this, many other edge points are detected, but some of them do not really belong to an edge. In this paper we use a new edge detector, that detects entire pieces of 1-pixel thick lines. Besides, this algorithm is more robust concerning the choice of the threshold value, because it works with the average gradient value of each piece of line. The proposed algorithm starts from pixels with a high gradient value, and follows neighbour pixels connecting new points.

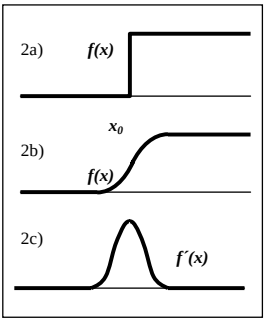


Fig. 2. An ideal edge

Let *f(x)* be an 1-dimensional signal, as shown in figure 2a), where there exists an ideal edge in the point x_0. Usually, in real sampled images, the edge is different to figure 2a), but it is similar to figure 2b). If we compute the gradient function, $g(x) = |\nabla f(x)|$, we obtain the curve of the figure 2c). As it can be seen, the edge can be detected by searching the maximum gradient value.

Let us see now a 2-dimensional image, $F(x,y)$, as shown in figure 4a), with an obvious edge that separates the dark and clear zones. Figure 4b) represents the gradient image, $G(x,y) = |\nabla F(x,y)|$. It can be seen that pixels with the highest value are precisely the edge points.

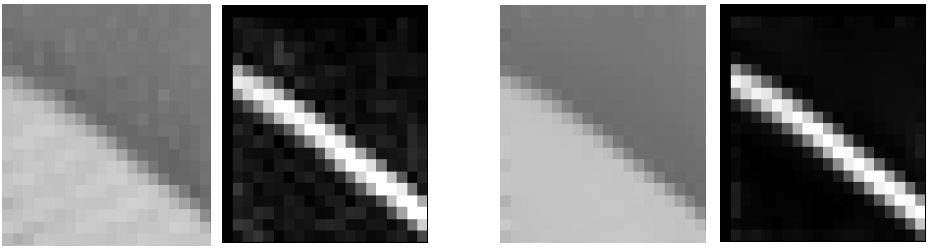


Fig. 3. On the left, edge image and its gradient image. On the right, the denoised image and its gradient image.

However, detecting points with a high gradient value is not enough to obtain the entire edge line. In some cases, the intensity values at both sides of the edge can be similar, producing, as a result, low gradient values, as shown in figure 4. Besides, another problem could be that in areas where the edge region is very wide, there are many points with a high gradient value, producing edge lines with more than 1-pixel thick.

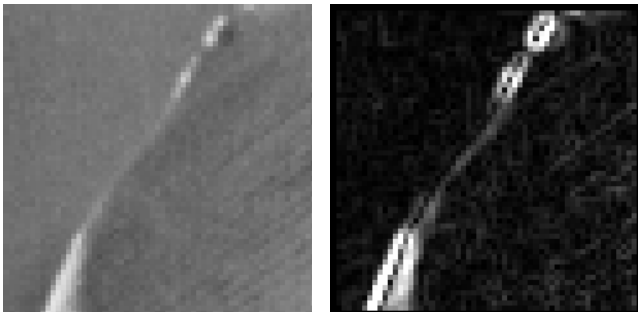


Fig. 4. Detail. The original image an its gradient image.

Our proposed algorithm works in a similar way to the contour-following algorithms described in [5]. It starts from a point, $P_0 = (x_0,y_0)$, with a very high value in the gradient image. The direction of the gradient vector at that point indicates the direction where the intensity variation in the image is larger. Besides, the high-level curves of the image function will be in the perpendicular direction of the gradient vector. The edge line that we are looking for does not really coincide with the high-level curve of the image function, except for ideal images where the intensity values are constant in both sides of the image, but its direction will be similar.

Let $Q(t)$ be the curve that characterizes the edge that we are looking for, and let $P_0 = Q(t_0) = (x_0,y_0)$ be a point of that curve. This point P_0 is a maximum extreme value of the function $F(x,y)$ in the direction $\nabla F(x0,y0)$, as can be seen in figure 5. And in addition to P_0, all the points $Q(t)$ are maximum extreme values of $F(x,y)$ in the direction $\nabla F(x,y)$. Therefore, in order to characterize the edge, we start from P_0 and keep following every point that verifies this property.

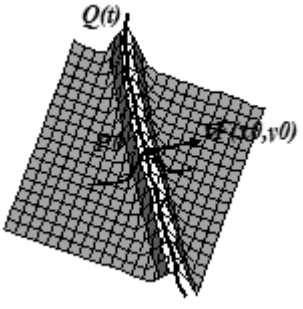


Fig. 5. 3D representation of the gradient where the pixel intensity values as heights in the *z* axis

From a geometric point of view, we can consider the pixel intensity values as heights in the *z* axis, and therefore consider the gradient image as a 3-dimensional surface. Then, what the algorithm is really doing is starting from a maximum height value on this surface (the top of a mountain), and keep following along the highest part of the 'chain of mountains' where we are positioned.

Working in this way, the algorithm can obtain its main goals. In the first place, the obtained lines are always 1-pixel thick, because the movement is always along the direction of the edge curve. In the second place, those areas with a small difference between intensity values (heights) in both sides of an edge, will be also included in the result line. This case occurs if the 'chain of mountains' where we are moving downwards until having a low height. If this height value does not coincide with the height of the base ground, the algorithm will work properly.

2.3 Quantification

The existence of noise in the image makes difficult the process of obtaining this window automatically. To improve the process a filter that eliminates the noise will be applied [1] [2]. Also a quantification filter will be applied to decrease the number of grey levels.

For it will be made it use of a defined quantifier in the following way:

The used quantifier associates to each image two finite sets:

$\{u_k\}_{k=1,..,S}$ that they are the final levels of grey in the image
$\{t_k\}_{k=1,..,S+1}$ that they are the separators of the quantification that satisfy:

$$-\infty < t_1 < u_1 < t_2 < ... < u_S < t_{S+1} < \infty.$$

The quantified image is generated replacing each value of the interval $[t_k, t_{k+1}]$ for u_k. The form in that it associates a quantifier to an image is through the minimization of an energy that determines the vicinity from the original image to the quantified image. The energy that is used is a variation of Max-Lloyd's classic Energy given by:

$$E(Q_S) = \sum_{k=1}^{S} \left(\left(\int_k^{k+1} (s-u_k)^2 \, dH(s) \right) + \frac{\lambda}{t_{k+1}-t_k} + C\left(\frac{t_{k+1}+t_k}{2} - u_k \right)^2 \right) \quad (2)$$

Where H(s) represents the function of distribution of the histogram h(s) of the image. Not it is desirable that the values t_k, t_{k+1} or u_k, u_{k+1} is too next, since visually would represent a single level, hence the terms are used:

$\frac{\lambda}{t_{k+1}-t_k}$ penalizes that intervals $(t_k, t_{k+1}]$ are small and $C\left(\frac{t_{k+1}+t_k}{2} - u_k \right)^2$ locates the grey level u_k in the center of $(t_k, t_{k+1}]$.

3 Results

We use in this work the Image Processing Environment XMW [3]. One of the inconveniences working with snakes is that the points of polygon must be introduced manually. It makes difficult to use of these techniques in a series of parallel images dedicated to the three-dimensional reconstruction.

Another problem is that usually the image contains a big amount of information, and the selected object may be small with regard to the image. In these conditions the snake method is not working, because this method takes information of the whole image.

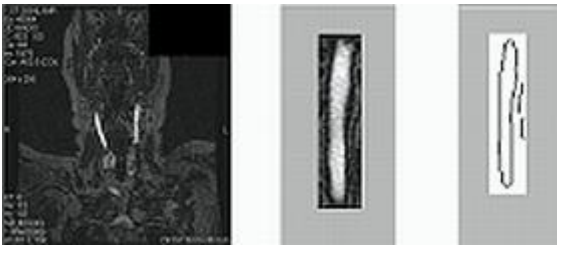


Fig. 6. In this figure we can see the window in the image (a), the subimage (b), and the result of the Thin Edges Filter (c)

Then a small subimage containing the object must be defined. This subimage should must be generated and located automatically starting from a only point selected by user inside the object to be detected.

In order to use these data in a three-dimensional reconstruction to eliminate the manual way to introduce the subimage in the original image it's necessary. Then, selecting an only point in the first image (image z_0) (see image 1) where the object

appears to be studied it is necessary to create the subimages in the different images where this same object appears.

In order to do this, we look for the center of the object that we want to detect and we initiate in the following image the search of the subwindow from that point. The process continues until the size of the subwindow is much bigger than the previous one.

The solution grows with a seed procedure, starting from the selected point, while the gradient of the points of the neighbour is smaller than a given threshold. The main goal is to obtain the maximum and minimum coordinates in both dimensions that are closer to the edge of the object. When the gradient of one neighbour be higher, the maximum and minimum coordinates in both dimensions (x_{max}, y_{max}, x_{min} , y_{min}) defines the corners of the minimum rectangle around the object. Then, the subimage is defined by $x_{max}+\varepsilon$, $y_{max}+\varepsilon$, $x_{min}-\varepsilon$ and $y_{min}-\varepsilon$ for a $\varepsilon>0$. (see Image 6b).

For the rest of images we start from a point in the same coordinates of the previous image and the previous procedure will be repeated, until the object disappears.

Once the image contains the object, the following step it is to apply a filter (Thin Edges) [5] to obtain an initial approach of the border of the object.

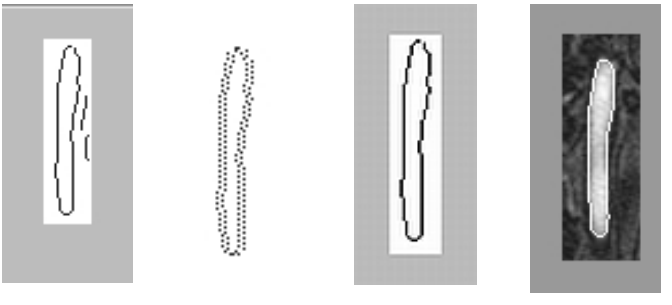


Fig. 7. In this figure we can see the result of the Thin Edges Filter (a), the result to order this points (b), the finally snake (c) and the snake in the original image (d)

The result of the edge filter is a Boolean image, where black pixels indicate border. This image must be processed to obtain an ordered list of connected points that define the whole border of the object. To do that, a Euclidean distance function will be used to insert connected points in the list, avoiding the non connected points (see figure 7b). That list will be the input data for the snake algorithm.

References

1. Alvarezy, L., Esclarín, J.: Image quantization using reaction-diffusion equations. SIAM Journal on Applied Mathematics 57(1), 153–175 (1997)
2. Alvarez, L., Esclarín, J., González, E., Mazorra, L.: Image Enhancement Using and Optimum Quantizer. In: Proceedings of the A Selection of Papers from the 6th International Workshop on Computer Aided Systems Theory (February 1997)
3. Trujillo, A., González, E.: XmegaWave, an Image Processing Environment. IEEE Computer Vision and Pattern Recognition Conference (CVPR 1998) (1998)

4. Computer vision and image processing in environmental research. Systems Analysis Modelling Simulation, 43(9), (2003)
5. Trujillo, A.: Thin edge detector. In: Proceeding of the X International Conferences on Image Analysis and Processing (ICIAP 1999) (September 1999)
6. Alemán-Flores, M., Alemán-Flores, P., Alvarez, L., Esteban-Sánchez, M.B., Fuentes-Pavón, R., Santana-Montesdeoca, J.M.: Semiautomatic Snake-based segmentation of solid breast nodules on ultrasonography. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 467–472. Springer, Heidelberg (2005)
7. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. International Journal of Computer Vision 22, 61–79 (1997)

Symmetric Optical Flow

L. Alvarez, C.A. Castaño, M. García, K. Krissian, L. Mazorra,
A. Salgado, and J. Sánchez

Departamento de Informática y Sistemas, Universidad de Las Palmas de Gran Canaria, Campus de Tafira s/n, 35017 Las Palmas de Gran Canaria, Spain
Tel.: +34 928 458708, Fax: +34 928 458711
{lalvarez,ccastano,mgarcia,krissian,lmazorra,
asalgado,jsanchez}@dis.ulpgc.es

Abstract. One of the main technique used to recover motion analysis from two images or to register them is variational optical flow, where the pixels of one image are matched to the pixels of the second image by minimizing an energy functional. In the standard formulation of variational optical flow, the estimated motion vector field depends on the reference image and is asymmetric. However, in most application the solution should be independent of the reference image. Only few symmetrical formulations of the optical flow has been proposed in the literature, where the solution is constraint to be symmetric using a combination of the flow in both directions. We propose a new symmetric variational formulation of the optical flow problem, where the flow is naturally symmetric. Results on the Yosemite sequence show an improved accuracy of our symmetric flow with respect to standard optical flow algorithm.

1 Introduction

The problem of motion analysis or registration between two images is an important problem that has been widely addressed in the literature. One of the main technique used to solve this problem is optical flow, where the pixels of one image are matched to the pixels of the second image. Hence, the estimated motion vector field depends on the reference image and is asymmetric. However, in most application the solution should be independent of the reference image. Symmetrical formulations of the optical flow has been proposed in [1,2,3], where the solution is constraint to be symmetric using a combination of the flow in both directions.

In [1], the consistency (or symmetry) of the mapping between two images I_1 and I_2 is enforced by jointly estimating the mapping from I_1 to I_2 and the mapping from I_2 to I_1 and by constraining those two mapping to be inverse to each other. The authors in [2] propose a different approach where the transformation between the two images, denoted T, is obtained by minimizing an energy functional such that $E(I_1, I_2, T) = E(I_2, I_1, T^{-1})$. To achieve this symmetric property, the energy is defined as the average of a non-symmetric energy applied to T and its inverse T^{-1}. Making the hypothesis that $E(T)$ can be described from the derivatives of T, the energy $E(T^{-1})$ is deduced from $E(T)$ without explicitly computing the inverse transformation T^{-1}. This approach is interesting but

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 676–683, 2007.
© Springer-Verlag Berlin Heidelberg 2007

leads to numerical difficulties when the derivatives of the transformation T are small. The approach in [3] is similar to the one of [1], but the authors explicitly take into account the discontinuities of the flow and the possible occlusions.

We propose a new symmetric variational formulation of the optical flow problem, where the flow is naturally symmetric. Results on the Yosemite sequence show an improved accuracy of our symmetric flow with respect to standard optical flow algorithm.

2 Formulation of the Problem

In the standard variational optical flow approach, given two images I_1 and I_2, the problem consists in finding a displacement image $\mathbf{u}$, where $I_1(\mathbf{x}) = I_2(\mathbf{x}+\mathbf{u})$. In order to find a displacement between 2 images I_1 and I_2 in a symmetric way, we consider an intermediate image I_m at half way between I_1 and I_2, so that there exists a displacement field $\mathbf{u}$ which fulfills $\forall \mathbf{x}, I_m(\mathbf{x}) = I_1(\mathbf{x} - \frac{\mathbf{u}}{2}) = I_2(\mathbf{x} + \frac{\mathbf{u}}{2})$. We illustrate this approach in Fig. 1.

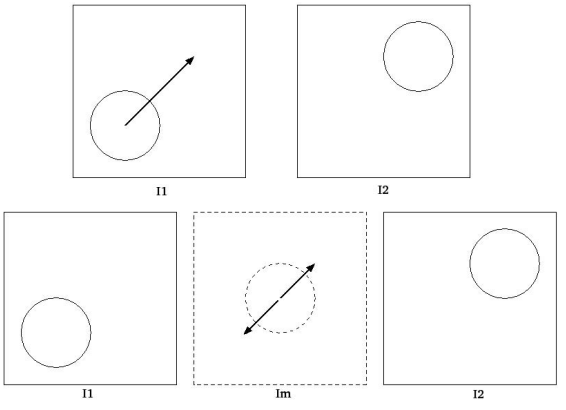


Fig. 1. Illustration of the hypothesis of the symmetric flow

To estimate this displacement, we minimize the energy:

$$E(\mathbf{u}) = \underbrace{\int_{\Omega} \left(I_1(\mathbf{x}^-) - I_2\left(\mathbf{x}^+\right)\right)^2 d\mathbf{x}}_{\text{data term}} + \alpha \underbrace{\int_{\Omega} \|\nabla \mathbf{u}(\mathbf{x})\|^2 d\mathbf{x}}_{\text{regularization term}}, \tag{1}$$

where we denote $\mathbf{x}^+ = \mathbf{u} + \frac{\mathbf{u}}{2}$, $\mathbf{x}^- = \mathbf{x} - \frac{\mathbf{u}}{2}$, and α is a scalar coefficient that weights the regularization (or smoothing) term. Under the assumption of intensity conservation for each voxel, the first term (*data term*) becomes zero when the first image matches the second one : $I_1(\mathbf{x}^-) = I_2(\mathbf{x}^+)$. This term tries to find the vector field that best fits the solution. The second term is a *regularization term* which smooths the vector field. A classification of different regularizers

can be found in [4]. In particular, the authors distinguish between image-driven and flow-driven regularizers and between isotropic and anisotropic ones. In this paper, since we are mainly interested in the symmetrical property of the flow, we simply use the L_2 norm presented above. The coefficient α is normalized to allow invariance under global intensity change of the form $(I_1, I_2) \rightarrow (k\ I_1, k\ I_2)$. To this purpose, α is multiplied by

$$\alpha = \alpha_0 \left(\epsilon + \sqrt{\frac{1}{|\Omega|} \int_\Omega \left\| \frac{\nabla \mathbf{I_1}(\mathbf{x}^-) + \nabla \mathbf{I_2}(\mathbf{x}^+)}{2} \right\|^2 d\mathbf{x}} \right)^2 \tag{2}$$

with $\epsilon = 10^{-3}$.

3 Temporal Regularization

We can use the image sequence information to impose temporal homogeneity and smoothness to the estimated flow. To this end, we re-formulate the equation, considering the whole sequence as a $2D{+}t$ image $I(\mathbf{y})$ where $\mathbf{y} = (\mathbf{x}, t) = (x, y, t)$. We define $\mathbf{y}^- = \mathbf{y} - \frac{\mathbf{u}(\mathbf{y})}{2}$, and $\mathbf{y}^+ = \mathbf{y} + \frac{\mathbf{u}(\mathbf{y})}{2}$, where the flow $\mathbf{u} = (u, v, dt)^t$ and dt is the constant time between two successive frames of the sequence.

The energy to minimize for the symmetric case is then written as:

$$E(\mathbf{u}) = \underbrace{\int_\Omega \left(I^-(\mathbf{y}) - I^+(\mathbf{y}) \right)^2 d\mathbf{y}}_{\text{data term}} + \alpha \underbrace{\int_\Omega (\mathbf{u}_x^2 + \mathbf{u}_y^2) d\mathbf{y}}_{\text{spatial reg.}} + \alpha_t \underbrace{\int_\Omega \mathbf{u}_t^2 d\mathbf{y}}_{\text{temporal reg.}}, \tag{3}$$

where $\mathbf{u}_x^2 = \frac{\partial u}{\partial x}^2 + \frac{\partial v}{\partial x}^2$, $\mathbf{u}_y^2 = \frac{\partial u}{\partial y}^2 + \frac{\partial v}{\partial y}^2$, $\mathbf{u}_t^2 = \frac{\partial u}{\partial t}^2 + \frac{\partial v}{\partial t}^2$ and α_t is a coefficient that weights the temporal regularization. Both coefficients α and α_t are normalized according to (2).

4 Implementation

To minimize the energies previously defined (without and with temporal regularization), we first calculate their gradients and we solve the corresponding Euler-Lagrange equations. In this section, we describe the non-temporal case from which the corresponding implementation of (3) can be easily deduced.

Euler-Lagrange equations from (1) yield:

$$(I_1(\mathbf{x}^-) - I_2(\mathbf{x}^+)).\frac{\nabla I_1(\mathbf{x}^-) + \nabla I_2(\mathbf{x}^+)}{2} + \alpha div(\nabla \mathbf{u}) = 0. \tag{4}$$

In order to linearize this equation, we use an iterative scheme where the displacement field $\mathbf{u}$ is successively estimated from the previous result:

$$\begin{cases} \mathbf{u}^0 & = \mathbf{u_0} \\ \mathbf{u}^{k+1} & = \mathbf{u}^k + \mathbf{h}^{k+1}, \end{cases} \tag{5}$$

where we update the vector field $\mathbf{u}$ at each iteration by adding another vector field $\mathbf{h}$. Supposing that $\mathbf{h}$ is small (which is true in practice since we use a

pyramidal approach) and neglecting the terms in spatial second order derivatives, we can write:

$$I_1(\mathbf{x}^-) \approx I_1(\mathbf{x} - \frac{\mathbf{u}^k}{2}) - \nabla I_1^t(\mathbf{x} - \frac{\mathbf{u}^k}{2})\frac{\mathbf{h}}{2} \tag{6}$$

$$I_2(\mathbf{x}^+) \approx I_2(\mathbf{x} + \frac{\mathbf{u}^k}{2}) + \nabla I_2^t(\mathbf{x} + \frac{\mathbf{u}^k}{2})\frac{\mathbf{h}}{2}. \tag{7}$$

Let us denote

$$\mathbf{g}(\mathbf{x}) = \frac{\nabla I_1(\mathbf{x} - \frac{\mathbf{u}^k}{2}) + \nabla I_2(\mathbf{x} + \frac{\mathbf{u}^k}{2})}{2} \tag{8}$$

$$d(\mathbf{x}) = I_1(\mathbf{x} - \frac{\mathbf{u}^k}{2}) - I_2(\mathbf{x} + \frac{\mathbf{u}^k}{2}). \tag{9}$$

The Euler-Lagrange equation (4) is then written as (at the current location $\mathbf{x}$):

$$(d\mathbf{g} + \alpha \Delta \mathbf{u}^k) - \mathbf{g}\mathbf{g}^t\mathbf{h} + \alpha \Delta \mathbf{h} = 0. \tag{10}$$

After discretization using finite differences, the Laplacian operator $\Delta\mathbf{h}$ can be divided in two terms $-2n\mathbf{h}$ and $S(\mathbf{h})$, where the n is the image dimension. The first term only depends on values of $\mathbf{h}$ at the current position $\mathbf{x}$ and the second term only depends on values of $\mathbf{h}$ at neighbor positions of $\mathbf{x}$: the vector $S(\mathbf{h})$ is written:

$$S(\mathbf{h}) = \begin{pmatrix} \sum_{\mathbf{p}\in N^*(\mathbf{x})} h^x(\mathbf{p}) \\ \sum_{\mathbf{p}\in N^*(\mathbf{x})} h^y(\mathbf{p}) \end{pmatrix}, \tag{11}$$

where $N^*(\mathbf{x})$ denotes the direct neighbors of $\mathbf{x}$ (4 in 2D and 6 in 3D), and $\mathbf{h} = (h^x, h^y)^t$.

Using $\mathbf{h}^{k+1}$ for the current location $\mathbf{x}$ and $\mathbf{h}^k$ for its neighbors, (10) becomes:

$$A\mathbf{h}^{k+1} = b, \tag{12}$$

with $A = \mathbf{g}\mathbf{g}^t + \alpha 2n\, I$, and $b = d\mathbf{g} + \alpha div(\nabla \mathbf{u}^k) + S(\mathbf{h}^k)$. The matrix A is real, symmetric and positive definite, so it can be inverted and we can compute for each position $\mathbf{x}$, $\mathbf{h}^{k+1} = A^{-1}b$. To improve the convergence rate, we use a Gauss-Seidel method which updates the displacement $\mathbf{h}^{k+1}$ at position $\mathbf{x}$ using the values of $\mathbf{h}^{k+1}$ already calculated. This scheme is recursive and to avoid privileging the direction of scanning the image, we apply two successive iterations of Gauss-Seidel in reverse directions. Furthermore, we use a pyramidal approach to compute the displacement flow at several scales, using the results from a given scale to initialize to the following higher scale.

5 Experiments and Results

In order to quantify the accuracy of our symmetric variational optical flow, we use the standard Yosemite sequence. The performance of the algorithm is measured using the angular error as described in [5]. If we consider velocities as 3D vectors were the third component has a constant value of 1, and if we denote

$\mathbf{u}_c = (u_c, v_c, 1)^t$ the correct velocity and $\mathbf{u}_e = (u_u, v_e, 1)^t$ the estimated one, the angular error is defined as:

$$\psi_E = arccos(\frac{\mathbf{u}_c \cdot \mathbf{u}_e}{\|\mathbf{u}_c\| \, \|\mathbf{u}_e\|}). \tag{13}$$

To compare our results with the ground truth, we have to transform the symmetric flow $\mathbf{u}$ into a standard flow $\mathbf{v}$ defined as $I_1(\mathbf{x}) = I_2(\mathbf{x} + \mathbf{v}(\mathbf{x}))$, according to $\mathbf{v}(\mathbf{x} - \frac{\mathbf{u}}{2}) = \mathbf{u}(\mathbf{x})$.

The flow $\mathbf{v}(\mathbf{x})$ is computed as a weighted average of the values of $\mathbf{v}(\mathbf{x} - \frac{\mathbf{u}}{2})$ in the neighborhood of $\mathbf{x}$. This transformation, described in appendix, is similar to the one proposed in [6]. In some sense, we can consider this transformation as a spatial inversion of the flow. To estimate the error introduced by this inversion, we applied it twice to the ground truth and compared the result to the initial ground truth. The angular error obtained between both flow was about 0.3 degrees which can be interpreted as a error of 0.15 degrees for a single inversion. However, this error can be reduced by upsampling the flow image before inversion and downsampling the result.

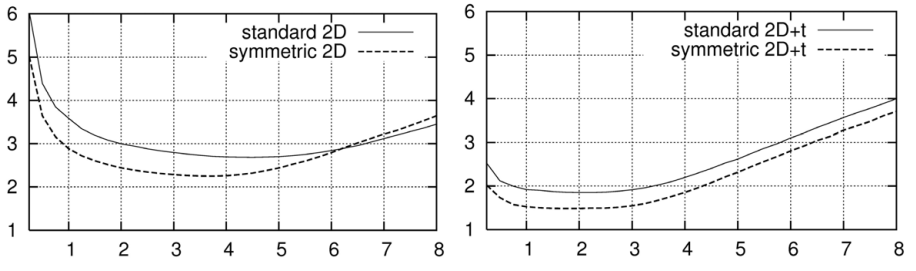


Fig. 2. Angular error obtained on Yosemite sequence, for the standard and the symmetric approaches to variational optical flow. Left: 2D algorithms, right: 2D+t algorithms including a temporal regularization of the flow.

In our experiments, we used the same parameters for the symmetric and the non-symmetric algorithms. The initial images have been smoothed by a Gaussian convolution of standard deviation 0.6 before the estimation, the image intensity at floating-point position was interpolated using a second-order spline interpolation and we used 3 pyramidal scales were the image dimensions were divided by two in X and Y directions from one scale to the next. For the temporal version, the coefficient α_t was set to 2 before its normalization. We applied a series of tests with different values of the regularization parameter α ranging from 0.25 to 8.0 with a step of 0.25. Results for both algorithms are depicted in Fig. 2. Despite the potential error introduced by the inversion of the flow, we observe a better behavior of the symmetrical version of the algorithm. The symmetric version of the optical flow reaches a better result both in the 2D and in the 2D+t cases, with minimal angular errors of 2.25 and 1.48 degrees respectively, while the standard approach reaches minimal angular errors of 2.68 and 1.85 degrees respectively.

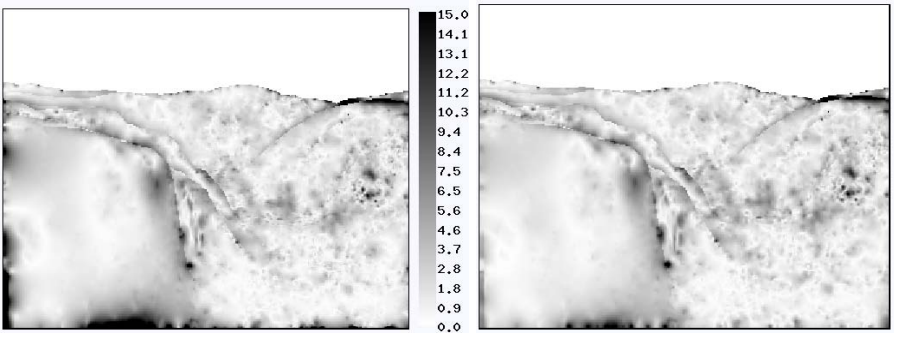


Fig. 3. Angular error obtained on the Yosemite sequence. Left, best result obtained for the 2D standard approach, right, best result obtained for the 2D symmetric approach.

As we can see in Fig. 3, most of the difference between the symmetric and the standard approach lies close to the image boundaries, mainly because the symmetric flow just needs half of the total displacement in each direction instead of the full vector. However, few differences of the angular error appear in the center of the image, where it is difficult of appreciate if an approach is better than the other.

6 Conclusion

In this paper, we proposed a new approach to variational symmetric optical flow. This approach has the advantage of simplicity over previously proposed approaches and it is similar to symmetric approaches used in cross-correlation techniques. We detailed our numerical scheme and we evaluated our approach on the standard Yosemite sequence using both a 2D and a 2D+t regularizations. Results show that the symmetrical version have a better behavior at the image borders leading to a improved mean angular error compared to a standard approach. In our future work, we plan to apply this new symmetric approach to more standard sequences used in optical flow, to experiment different regularizers like the Nagel-Enkelmann regularizer, and to apply this approach in the context of 2D and 3D Particle Image Velocity images.

Acknowledgments

Founded by the European Project FLUID (contract no. FP6-513663).

References

1. Christensen, G., Johnson, H.: Consistent image registration. IEEE Transactions on Medical Imaging 20(7), 568–582 (2001)
2. Cachier, P., Rey, D.: Symmetrization of the non-rigid registration problem using inversion-invariant energies: Application to multiple sclerosis. In: Delp, S.L., DiGoia, A.M., Jaramaz, B. (eds.) MICCAI 2000. LNCS, vol. 1935, pp. 472–481. Springer, Heidelberg (2000)

3. Alvarez, L., Deriche, D., Papadopoulo, T., Sánchez, J.: Symmetrical dense optical flow estimation with occlusions detection. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 721–735. Springer, Heidelberg (2002)
4. Weickert, J., Schnörr, C.: A theoretical framework for convex regularizers in pde-based computation of image motion. International Journal of Computer Vision 45(3), 245–264 (2001)
5. Barron, J., Fleet, D., Beauchemin, S.: Performance of optical flow techniques. IJCV 12(1), 43–77 (1994)
6. Salgado, A., Sánchez, J.: Optical flow estimation with large displacements: A temporal regularizer. Technical report, Instituto Universitario de Ciencias Tecnology

Computing the Backward Flow, v

In this appendix we examine how to compute the flow from I_1 to I_2 from the symmetric flow. The correspondence between both flows is

$$\mathbf{u}(\mathbf{x}) = \mathbf{v}\left(\mathbf{x} - \frac{\mathbf{u}}{2}\right). \tag{14}$$

The main difficulty here is to deal with discrete images. The displacement flow $\mathbf{v}$ that we are looking for takes values at pixel location, but we only have its values at the locations $\mathbf{x} - \frac{\mathbf{u}}{2}$, which are not centered on pixels in general. We consider each pixel as a square. As we can see in Fig. 4 we have to adjust the value of $\mathbf{v}(\mathbf{x})$ depending on the portion of pixels that arrive into the pixel $\mathbf{x}$. In general, there will be several correspondences that distribute their values on a single pixel, so we propose to compute an average of all the portions of the flow that fall into each pixel.

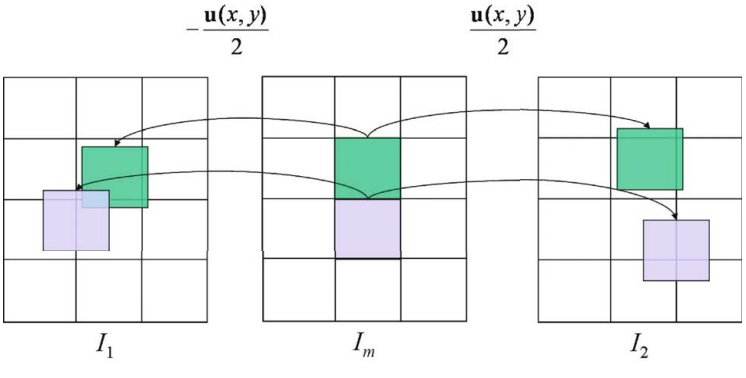


Fig. 4. Estimation the flow from I_1 to I_2 given the symmetric flow. As we can see in this figure, we have to divide each correspondence in four different estimates –four different pixels– in order to compute the values of the discrete function $\mathbf{v}(\mathbf{x})$.

Considering the discrete nature of $\mathbf{v}(\mathbf{x})$ we may compute its value in each position according to this functional

$$\mathbf{v}(\mathbf{x}_i) = - \frac{\sum_{j=1}^{N} \mathbf{u}(\mathbf{x}_j) p_{i,j} \left(\mathbf{x}_i, \mathbf{x}_j - \frac{\mathbf{u}(\mathbf{x}_j)}{2} \right)}{\sum_{j=1}^{N} p_{i,j} \left(\mathbf{x}_i, \mathbf{x}_j - \frac{\mathbf{u}(\mathbf{x}_j)}{2} \right)},$$

where N is the size of the image and $p_{i,j}$ stands for the area of the pixel j that fall into pixel i as can be seen in Fig. 4. Each $\mathbf{x}_j - \frac{\mathbf{u}(\mathbf{x}_j)}{2})$, we generate four different $p_{i,j}$ that will lie on neighbouring pixels. Lets see how we compute these weights. To simplify we call $\mathbf{a} = (a_x, a_y) = \mathbf{x}_i$ and $\mathbf{b} = (b_x, b_y) = \mathbf{x}_j - \frac{\mathbf{u}(\mathbf{x}_j)}{2}$

$$p_{i,j}(\mathbf{a}, \mathbf{b}) = \begin{cases} 0 & if\ max(|a_x - b_x|, |a_y - b_y|) \geq 1 \\ (1 - (b_x - a_x)) \cdot (1 - (b_y - a_y)) & if\ a_x < b_x\ and\ a_y < b_y \\ (1 - (b_x - a_x)) \cdot (1 + b_y - a_y) & if\ a_x < b_x\ and\ a_y > b_y \\ (1 + b_x - a_x) \cdot (1 - (b_y - a_y)) & if\ a_x > b_x\ and\ a_y < b_y \\ (1 + b_x - a_x) \cdot (1 + b_y - a_y) & if\ a_x > b_x\ and\ a_y > b_y \end{cases}.$$

A further issue to consider are the empty pixels,$\mathbf{x}_i$, that have no correspondence in the other image –usually due to occlusions–. Normally these pixels are situated close to the object boundaries in the direction of their displacements. For these reasons we apply a post–processing step in order to fill up the holes. This is a simple step in where after several iterations we complete the information of the holes by averaging with the information from the neighbours.

Real-Time Stereo Visual SLAM in Large-Scale Environments Based on SIFT Fingerprints

David Schleicher, Luis M. Bergasa, Manuel Ocaña,
Rafael Barea, and Elena López

Department of Electronics, University of Alcala, Alcalá de Henares, 28805 Madrid, Spain
dsg68818@telefonica.net,
{bergasa,barea,elena,mocana}@depeca.uah.es

Abstract. This paper presents a new method for real-time SLAM calculation applied to autonomous robot navigation in large-scale environments without restrictions. It is exclusively based on the visual information provided by a cheap wide-angle stereo camera. Our approach divide the global map into local sub-maps identified by the so-called SIFT fingerprint. At the sub-map level (low level SLAM), 3D sequential mapping of natural land-marks and the robot location/orientation are obtained using a top-down Bayesian method to model the dynamic behavior. A high abstraction level to reduce the global accumulated drift, keeping real-time constraints, has been added (high level SLAM). This uses a correction method based on the SIFT fingerprints taking for each sub-map. A comparison of the low SLAM level using our method and SIFT features has been carried out. Some experimental results using a real large environment are presented.

Keywords: SLAM, Intelligent Vehicles, Computer Vision, Real-Time.

1 Introduction

Real-time Simultaneous Localization and Mapping (SLAM) is a key component in robotics. In the last years several approaches have been used [1][2]. Recent researches have demonstrated that camera-based SLAM is very useful in domains where the goal is to recover 3D camera position in real-time moving rapidly in normal human environments, based on mapping of sparse visual features, potentially with minimal information about motion dynamics [3]. In [4], a 3D visual SLAM method, based on a stereo camera and SIFT (*Scale Invariant Feature Transform*) features, is presented. Currently, the main goal in SLAM research is to apply consistent, robust and efficient methods for large-scale environments. One of the main milestones is to achieve large closing loops in robot paths running in real-time.

Several approaches can be found to solve the related issues by using *metric* methods, *topological* methods or hybrid ones. One example of the last ones is described in [5]. This solution tries to build a topological map composed by several simple metric maps. After that, as long as the robot explores new places, the algorithm decides whether to build a new sub-map or create a new *branch* to one of the already visited maps. The links contain the coordinate system relations as well as uncertainties transformations among sub-maps. In [6] they propose a 3D SLAM using

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 684–691, 2007.
© Springer-Verlag Berlin Heidelberg 2007

SIFT similarity matrix, which are based on visual appearance. This allows the recognition of pre-visited places that can be quite repetitive. The paper presented in [7] proposes a method that is able to close a very large loop with a very high number of landmarks on a simulated environment. It uses a hierarchical method to represent the different probabilistic magnitudes associated to several map regions. Also, means to transfer the updates and predictions from the top to the bottom of the tree and vice versa are provided.

This paper presents a real-time SLAM method based on stereo-vision. The basis of this work was previously presented by the authors in [8]. The system is based on a stereo wide-angle camera mounted on a mobile robot. Several visual landmarks are sequentially captured, using the Shi and Tomasi operator (see [8]), and introduced on an EKF filter in order to model the probabilistic behavior of the system. A measurement model is used for landmarks perception and a motion model is implemented for the dynamic behavior of the robot. As it is well known, one of the major problems on the EKF implementation is the quadratic (n^2) increase of computational cost as a function of the number of landmarks, making it unsuitable for large environments where this number can be potentially high. In order to solve this problem, we present a modified SLAM implementation, which adds an additional processing level that we will call "high level SLAM" to the explained SLAM method, which will be known as "low level SLAM". The global map is divided into local sub-maps identified by SIFT fingerprints. A fingerprint characterize the visual appearance of an image from some SIFT visual landmarks and the relation among them. Then, the robot is locally positioned in a sub-map using the low level SLAM. The fingerprints are periodically taken along the path covered by the robot. This process identifies a sub-map as the landmark positions obtained from the images taken from the current *fingerprint* to the next one. A *fingerprint* matching between previously captured fingerprints and current one is carried out to detect pre-visited zones. The information extracted from the matched SIFT features of the *fingerprint* is used to update the EKF filter and correct the robot state as well as the whole map covered by the loop.

2 Low Level SLAM

This level implements all the algorithms and tasks needed to locate and map the robot on its local sub-map, which is detailed in [8]. In order to apply the EKF, a state vector X and its covariance matrix P need to be defined. Respect to the *measurement model*, visual measurements are obtained from the "visible" features positions. In our system we define each individual *measurement prediction* vector $h_i = \begin{pmatrix} h_{ix} & h_{iy} & h_{iz} \end{pmatrix}^T$ as the corresponding 3D feature position relative to the camera frame. Regarding the *motion model*, it is based on a more general application (see [3]), where the so-called *impulse model* is implemented. It has been adapted to the navigation of a mobile robot by using some restrictions associated to its movement.

3 High Level SLAM

As it was stated before, in large environments, as the number of landmarks grows the covariance matrix P size increases until the processing time exceeds real time

constraints. To avoid this, only a local visible window of landmarks is introduced into the EKF. This, as we will show on the results, allows an almost constant processing time with number of landmarks increasing (see Fig. 2 *Left-down*). On the other hand, we need to keep the global map error as low as possible. Therefore we need to assure the whole map consistency as well. As we only keep local uncertainty information on the visible features, it is necessary to add a higher level process that preserves the global map uncertainty history along the robot's path, that is the cumulative covariance matrix $P_G(k)=P_G(k-1)+P_{XX}(k)$. To do that we divide the global map into local sub-maps identified by the *SIFT fingerprints* $SF=\{sf_l | l \in 0...L\}$, each of them composed by a set of *SIFT features* $YF^l=\{Yf_m^l | m \in 0...M\}$. The process consists in periodically take fingerprints along the path covered by the robot. Each time a new fingerprint is to be taken it is evaluated comparing it to the previously acquired fingerprints within an uncertainty search region. This region is obtained from P_G because it keeps the global uncertainty information of the whole map. In case that the result of the evaluation gives that the robot is in a previously visited place, a *closed loop* situation will be identified. This situation is explained later on. The philosophy of the EKF sub-maps was previously presented in [12]. One of the main differences between their approach and ours is the local map management. In [12], the EKF process is carried out taking into account all landmarks within the local maps. We take into account only the visible ones. This allows us to keep always a reduced number of landmarks being processed within the EKF. On the other hand, in [12], a global map is tried to be built joining all local maps. Then, a process is carried out to identify all duplicated landmarks, closing all possible loops inside. Instead, we continuously keep the accumulated global uncertainty, allowing the global map correction at any time as soon as a closed loop situation is detected.

3.1 SIFT Fingerprints

As we explained before, the way to identify a place is based on the so-called SIFT fingerprint. These fingerprints are composed by a number of SIFT landmarks distributed across the reference image and characterize the visual appearance of the image. SIFT features are invariant to image scaling and rotation, and partially invariant to change in illumination and 3D camera viewpoint. In addition, the features are highly distinctive, which allows a single feature to be correctly matched with high probability. This is achieved by the association of a 128 length descriptor to each of the features, which will identify uniquely all of them. The detailed SIFT features extraction process is described in [9]. In our global process we will store these SIFT feature descriptors $\vec{\delta}$ within the global database, using them for the fingerprints matching process. The left image coordinates and the 3D position are also stored $Yf_m^l=\left(u_L \quad v_L \quad X \quad Y \quad Z \quad \vec{\delta}\right)$.

3.2 Matching Process

One of the main issues on SLAM in large environments is the *loop-closing* problem. The first issue to solve is the recognition of previously visited places, as it was stated before. Once a new fingerprint is identified it is evaluated, that is, it is compared

against all stored fingerprints within the uncertainty area. This comparison is carried out through a matching process which takes into account, for each pair of fingerprints (sf_A, sf_B), both the number of recognized SIFT features and their relative positions within the images to compare. The overall process is as follows: 1.) Computation of the *euclidean distance* between all detected SIFT features on both images, and selection of those close enough. 2.) Lines connecting each pair of matched features are calculated. The corresponding lengths $Ln_{i,j}^{A-B}$ and slopes $Sp_{i,j}^{A-B}$ are computed as well (see Fig. 2 *Right-down*). 3.) Outlayer features are excluded from the computation by the use of the *RANSAC* method. 4.) The global *fingerprint matching probability* is computed as a weighted function of 2 parameters: *Number of matched features* and *Inliers/Outliers relation* (see (1)).

$$P_{fp_match} = m_1 \cdot num_matches + m_2 (n_I / n_O) \cdot \tag{1}$$

Once the loop-closing situation has been detected, the whole map must be corrected according to the old place recognized. The first step is then, to update the current robot state $(X_{rob} \quad q_{rob} \quad v_{rob} \quad \omega)^T$ with the detected pre-visited place. To do that we use the epipolar geometry applied to the matched SIFT features in the same way as in the low level SLAM stage. This is achieved thanks to the stored fingerprint states X_{fp}, which represent the robot states at the time of the fingerprint creation. After that, the rest of the map, including feature positions Y_i and fingerprint states X_{fp}, along the loop must be updated accordingly. However, we must assure that the resultant map is *consistent*. The idea behind is that the global uncertainty P_G will always grow as long as the robot moves on the environment. That means: the error in self-locating will increase until the robot revisits an old place that helps to reduce its own uncertainty. In terms of map construction, we can conclude that the oldest map features will have less associated uncertainty and will have to be corrected in a lower degree. On the opposite side, a higher degree of correction will be applied to more recent features. This degree of correction will be modulated as a function of the historic accumulated uncertainty along the whole loop: P_G.

On equations (2), (3) and (4) we show the 3 consecutive steps applied to the position of a single feature in order to calculate its estimated new corrected value. $\mathrm{T}(P_{Gi})$ represents the trace of the global covariance of fingerprint associated to the feature i. $\vec{D}_R$ is the 3 components rotation expression of the robot rotation matrix R_{rob}. Finally, *Rot* represents the transformation of the 3 components vector to the rotation matrix expression. The attributes *init* and *end* refer to the situation before and after the map correction.

$$Y'_i = R_{init}^{-1} (Y_i^{init} - X_{init}) \cdot \tag{2}$$

$$Y''_i = Y'_i \cdot Rot\left[\frac{\mathrm{T}(P_{Gi})}{\mathrm{T}(P_{Gend})} \cdot \left(\vec{D}_{R\,fin}\right)^{-1} + \left(1 - \frac{\mathrm{T}(P_{Gi})}{\mathrm{T}(P_{Gend})}\right) \cdot \left(\vec{D}_{R\,init}\right)^{-1} \right] \cdot \tag{3}$$

$$Y_i^{fin} = Y''_i - \left[\frac{\mathrm{T}(P_{Gi})}{\mathrm{T}(P_{Gend})} \cdot X_{fin}^{-1} + \left(1 - \frac{\mathrm{T}(P_{Gi})}{\mathrm{T}(P_{Gend})}\right) \cdot X_{init}^{-1} \right] \cdot \tag{4}$$

Once the loop is closed, P_G takes the value of the associated old fingerprint identified. Thus, we update the global uncertainty to the new situation. In addition, old visited features will become visible again, and can be incorporated to the low level EKF process.

4 Results

For testing the behaviour of our system, a test video sequence has been used. The cameras employed were the Unibrain Fire-i IEEE1394 modules with additional wide-angle lens, which provide a field of view of around 100º horizontal and vertical. Both cameras are synchronized at the time of commanding the start of transmission. The calibration is performed offline using a chessboard panel using the method referenced in [10]. The test video sequence was taken by moving the robot along the upper floor of our Polytechnic School building. The complete path, from the start point to the loop-closing place, has a perimeter of 283.25 m. We have implemented the low level SLAM using two techniques: our proposal and the SIFT method introduced by D. Lowe [9], in order to compare its performance in a large environment. Fig. 1 *Left* depicts the obtained results based on the low level SLAM. Both ours and SIFT results are showed together with the ground truth data. From these results we observe two main deviation points on each implementation. The first deviation is accused with the SIFT method on the first curve of the path, while using our implementation, the robot tend to deviate on the third curve. If we represent the cumulative mean error ($\varepsilon_n = (1/n_{total}) \cdot \sum_{i=0}^{n} |X_i - Xref_i|$) on Fig. 1 *Right*, we appreciate that it appears to be higher on SIFT implementation. The reason is that the first deviation causes a cumulative drift over the rest of the path. Respecting to the high level SLAM, Fig. 2 *Up* depicts the representation of the map estimated, including all landmarks and fingerprints. The sequence represents the map right before and after the loop-closing situation. As it is

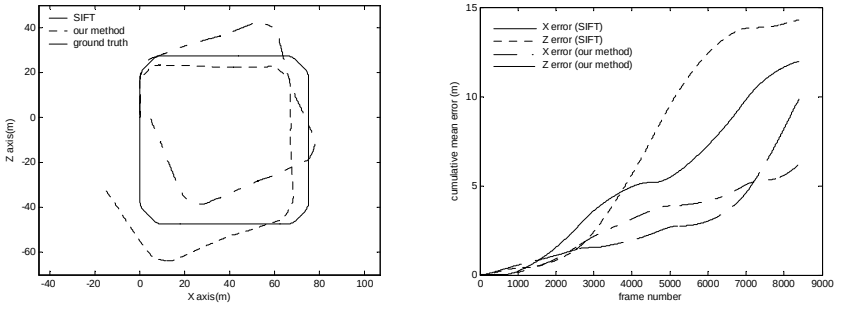


Fig. 1. *Left:* Estimation of the path covered by the robot using SIFT and our method. The reference (ground truth) is drawn on solid line. *Right:* Cumulative mean error for X axis and Z axis respect to the frame number n. Results are shown both for the SIFT implementation and our method.

shown, the map keeps the consistency also after its correction. After that situation, the system keeps the consistency re-introducing the old visible landmarks on the low level EKF and still detecting old fingerprints. Respecting the processing time, the real-time implementation imposes a time restriction, which shall not exceed 33 ms for a 30 frames/second capturing rate.

All results were taken using a 2.0 GHz speed CPU. Fig. 2 *Left-down* depicts the processing times along the whole robot path for both SIFT and our implementation.

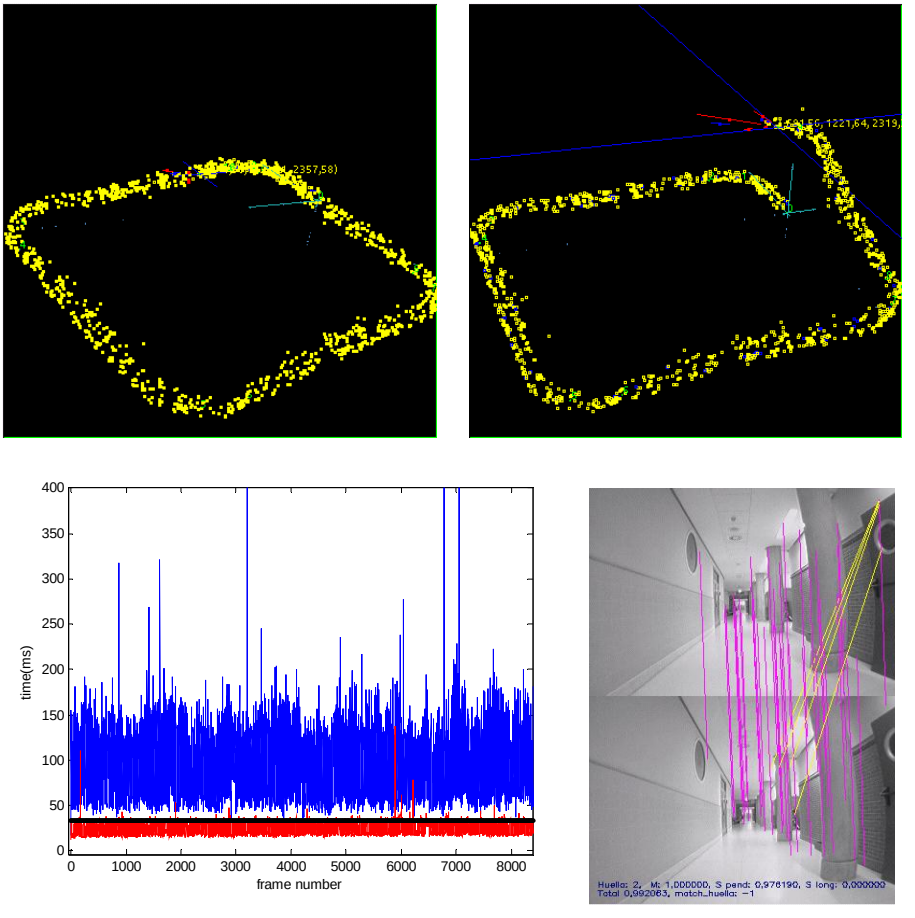


Fig. 2. Map representation of a loop-closing situation (3D graph). Low level landmarks are represented in yellow colour, while visible and correctly measured are represented in red colour. Green numbers show the fingerprint locations. The robot position is represented on the centre of the big blue cross: P_G. *Left-up:* Representation right before the map correction. *Right-up:* Representation right after the map correction. *Left-down:* Processing times for low level SIFT (in blue colour) and our method in red colour. Real time limit is represented as a constant 33 ms black line. *Right-down:* Fingerprint SIFT features matching. Inliers are shown on magenta colour while outliers are shown on yellow colour.

The processing time results are lower applying our method than using SIFT. We can also see that only our method is able to work under the real time constraint. Another conclusion we can take from the graph is that the average processing time remains quite constant along the whole path, showing the benefit of using our method instead of the SIFT one. We divided it into tasks associated to the low level SLAM and the high level SLAM. Respect to the first one, the average processing times using our method are 3ms for *measurements*, 5ms for *filter updates* and 7ms for *feature initializations*. Respect to the use of SIFT landmarks, the processing times are 47ms for *measurements*, 5ms for *filter updates* and 62ms for *feature initializations*. We can see the main cause of the higher processing time for SIFT implementation, which is due to the increase of *measurements* and *feature initialization* phases computational costs. Even though we restricted the keypoints search to the minimum needed area, the successive Gaussian blurring phases contribute to increase the processing time. This is particularly evident for the case of the feature initialization phase, where the search area is extended along the whole epipolar line, though we restricted its length for $1\text{m} \rightarrow \infty$ search range. Regarding the high level SLAM, it shows that the map is composed of a number of 1630 landmarks at the time of loop closing. The *Fingerprint matches* process needs 3s average time while the *loop closing* time was 4s. As it is shown, the time dedicated to *fingerprint matching* process is significantly higher than real time. This is also applicable to the correction of the map at the time of loop closing. It has to be taken into account that both tasks do not belong to the continuous self-locating process that is carried out by the low level SLAM. That means that there is no need to complete them within a single frame time slot. So, we can obtain a positive fingerprint matching result some few frames after it was really detected. Then, we can go back and start loop-closing task including also the last processed frames. This implies that both of these tasks can be computed in *parallel*, keeping them outside the real time computation.

5 Conclusion

We have presented a two hierarchical SLAM levels method that allows self-locating a robot by measuring the 3D positions of different natural landmarks. Several benefits have been shown on the low level SLAM respect to the use of SIFT features mainly on the processing time area. Respecting the high level SLAM based on SIFT fingerprints, it has been proved that it solves the loop-closing problem keeping the real time behavior constant along the path. Nowadays we are working on improving our taking fingerprints technique.

Acknowledgements

The authors would like to express their gratitude to the Comunidad de Madrid and the University of Alcalá for their support through the projects RoboCity2030 (CAM-S-0505/DPI/000176) and LOMUCO (CCG06-UAH/DPI-0721).

References

1. Lopez, E., Bergasa, L.M., Barea, R., Escudero, M.: A Navigation System for Assistant Robots Using Visually Augmented POMDPs. Autonomous Robots 19(1), 67–87 (2005)
2. Newman, P.M., Leonard, J.J., Neira, J., Tardós, J.: Explore and return: Experimental validation of real time concurrent mapping and localization. In: Proceedings of the IEEE Conference on Robotics and Automation, pp. 1802–1809. IEEE Computer Society Press, Los Alamitos (2002)
3. Davison, A.J.: Real-time simultaneous localisation and mapping with a single camera. In: Proceedings of the 9th International Conference on Computer Vision, Nice (2003)
4. Elinas, P., Sim, R., Little, J.J.: SLAM: Stereo Vision SLAM Using the Rao-Blackwellised Particle Filter and a Novel Mixture Proposal Distribution. In: ICRA (2006)
5. Se, S., Lowe, D., Little, J.: Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. The International Journal of Robotics Research 21(8), 735–758 (2002)
6. Newman, P., Cole, D., Ho, K.: Outdoor SLAM using Visual Appearance and Laser Ranging. In: ICRA (2006)
7. Frese, U., Schroeder, L.: Closing a Million-Landmarks Loop. In: IROS (2006)
8. Schleicher, D., Bergasa, L.M., Lopez, E., Ocaña, M.: Real-Time Simultaneous Localization and Mapping using a Wide-Angle Stereo Camera and Adaptive Patches. In: IROS2006
9. Lowe, D.G., Little, J.: Vision-based mobile robot localization and mapping using scale-invariant features. In: International Conference on Robotics and Automation, Seoul, Korea, pp. 2051–2058 (2001)
10. Heikkila Silven: A Four-step Camera Calibration Procedure with Implicit Image Correction. In: CVPR1997
11. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
12. Tardós, J., Neira, J., Newman, P., Leonard, J.: Robust mapping and localization in indoor environments using sonar data. Int. J. Robotics Research 21(4), 311–330 (2002)
13. Davison, A.J.: Mobile Robot Navigation Using Active Vision. PhD Thesis, University of Oxford (1998)

Tool for Creation Realistic Animation of Human-Like Figures: TRAF

Klempous Ryszard

Wroclaw University of Technology,
Institute of Computer Engineering, Control and Robotics 27 Wybrzeze Wyspianskiego , 50 - 370 Wroclaw, Poland
ryszard.klempous@pwr.wroc.pl

Abstract. Animation of human figures can be a complicated task, however its practical applications in various domains are gaining on popularity. One of the most common applications is creation of special effects in animation movies, computer games, sport and various virtual reality solutions. Motion capture is perhaps of the most intensively explored and fastest growing discipline. It allows for computer analysis of animated figures movements recorded in 3D space. Movement animation can be generated by recording human movements as well as artificially by a computer algorithm. The analysis of the pre-recorded movements plays an important role in practical applications such as biometric identification systems and medical information systems.

1 Introduction

Motion capture [1,10] is a technology that allows for interception and digital recording of human motions using highly specialised equipment and getting the data set so as to use for various applications. There are several useful technologies available for the task of motion registration, majority of these technologies relay on tracking markers placed on the actor's body. In general, the motions capture software system first records and then process various positions, velocities, accelerations, angles and impulses and thus providing an accurate digital representation of the motion. Such a general scheme for interception of a human movement sequence using the motion capture is demonstrated in Figure 1. In order to be able to register movement data we need to attach special markers to the joints of the actor's body. The actor then performs specified sequences of movements in a recording studio. First the original movements are captured and store (Fig.1a and Fig.1b) and then the obtained information about trajectories produced by sensors is stored in digital form (Fig.1c). The collected data can be used for re-generation on a computer display of human movement animation in 3D (Fig.1d).

The technique provides satisfying results in a form of human figures natural movements. The major obstacles in this approach can be a high cost of recording equipment and elaborate processes involved in movement registration. When creating animations based on motion capture technique we use only the previously

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 692–700, 2007.
© Springer-Verlag Berlin Heidelberg 2007

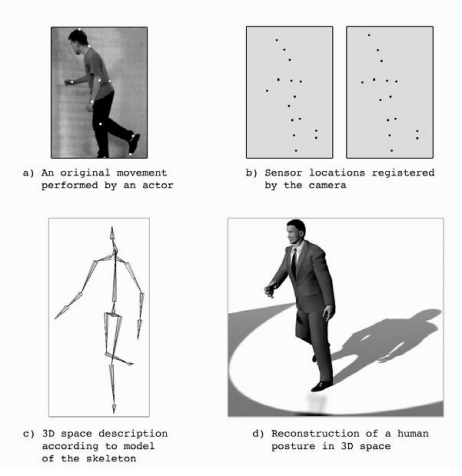


Fig. 1. Tool called TRAF (Tool for Creation of Realistic Animations of Human-like Figures

stored and accessible movement sequences represented by by frames movement vector.

To generate yet another animation that represents other types of movement we need to stage another session with human actors in the recording studio. At present the main subject of our research is investigation of various animation methods and related motion signal processing techniques that would allow multiple re-use and possible transformations of motion capture data that is already presorted in the database. Most common methods in use rely either on fitting the movements of one actor to that of another actor popular [5] or on joining movements two (or more movements) applying a smooth transition mechanism between them [3,8]. More advanced techniques involve extractions of movement types (styles) and injecting them into other movements [4]. Other areas involve linking motion types with representations of emotions [5,6,7]. Currently available tools for automatic creation of movement animation is still of limited capability and are characterised by their rather poor quality of input data. Considering all these problems and limitations when creating animations of more complex scenarios we came with the idea of creating a Database of elementary motions that can be shared among researchers in the domain of motion capture technology [8]. The main goal of automatic motion synthesis, in which a computer generates the entire animation, is described in [3]. In the [19] introduced a concept of motion texture as a set of motion textons and their distributions. System proposed in [19] allowed a reconstruction of a variety of motions statistically derived from an original dataset. An animator can specify key frames and then using a statistical model drawn from motion capture data to texture the key framed motion with a particular style thanks to [9,11,13] work. In that approach textons are modeled by a linear dynamic system and characterize the stochastic and dynamic nature of the captured motions[12,21]. Recent research efforts have demonstrated

approaches similar to ours in sense that they can retain the original motion data for use in synthesis. Arikan and Forsyth [2] search hierarchical motion graphs to generate desired motion in the real time. This paper is organized in a following manner. The key idea of TRAF is presented in the second section. In the third section a short review of methods for motion processing is presented including models and the motion representations. Some experiments that have been carried out and their illustrations are presented in fourth section. The last section consists of discussion and conclusions.

2 The Overview of TRAF

The creation of TRAF project was an attempt to construct a useful tool for automatic generation of realistic animation of human figure movements. The main was the process of segmentation achieved thanks to motion captures technique in which continuous movement was divided into elementary and uniform movement units that can be stored in a dedicated motion a Data Base (mDB). In this case the uniformity relates to the character of a specific movement here understood as related to a type of animated figure behaviour (activity) i.e. a stroll, run or a push-up on the floor. All collected elementary motions are then used to build much longer sequences of movements according to limitations or constraints set by the user of the system. Thanks to specific mechanisms of adjustment (fitting) and synthesis of elementary (atomic) motions the systems is able to generate very realistic animation an automatic manner. The realization of the TRAF project was initiated in collaboration with Jakub Segen from Multimedia Communications Research Laboratory, Bell Laboratories – Lucent Technologies, Murray Hill, NJ, USA. The novelty that distinguishes the TRAF toolkit from other similar applications is its ability to facilitate the re-use (re-targeting) of previously stored and specially processed sequences of motion data for production of various human movements' scenarios. Thanks to a dedicated mechanism of segmentation, from pre-stored sequences of motion capture only specific fragments [9], which after special ordering in the DB of the system can be used gain for a generation of new movements.

To create a new motion we have to deliver the following dedicated methods and tools for motion synthesis [13] as well as ensure the intercommunication between the system and the user [10]. Considering various aspects of data processing the system has been built from four (4) main functional modules as indicated in the Figure 2. Such approach to the problem of motion processing ensures an ease of use at each stage of the process. The modules of the TRAF system were divided in to two (2) functional groups: a.) Analysis of Motion Module; b.) Synthesis of Motion Module.

Each of these modules has first to process the information and then transfer the results to the rest of the modules of the system. The diagram in Figure 2 depicts the flow of data transfer in the TRAF.

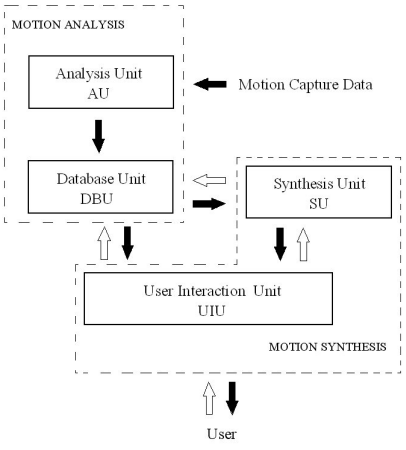


Fig. 2. High Level Model of the TRAF System

Motion Analysis Module

Its function is to process the input data – movements that are registered using motion capture technology.

These movements are to be used as a raw data. The raw data are analysed from the point of their elementary motion content. Selected captures of motions are transferred to the motion Data Base Module.

Motion Data Base Module stores motion data extracted from the raw data in the Motion Analysis Module.

Data is organised and stored in this module using specific structures. The module allows access to the pre-store movement data according to various requests originating from the Motion Synthesis or User Interaction Modules.

User Interaction Module role of this module is collect and process the user requests pertaining the system functions. The processed requests are passed on as constraint parameters to the Motion Synthesis Module. Direct interaction with the DB allows the users to review the content of the stored motion data records.

Motion Synthesis Module, after initial analysis of the limitation parameters received from the User Interaction Module the Motion Synthesis generates requests for data that are needed for motion synthesis. After the process of synthesis is completed the resultant animation is passed onto User Interaction Module for presentation to the user [8].

3 Important Issues Releated to Processing of Movement

Analysis is the key operation that is executed on the given sample of a motion. It is base on certain division of motion capture sequences into smaller chunks that are represented by the Primitive Motions. These motions are atomic (non-divisible) elements that are extracted from the whole motion capture sequence.

Prior to commencing the analysis motion data are interpreted and normalized using specialised input filters.

3.1 Analysis of Movement Data

The role of input filters is to normalise the figure's exoskeleton, as well as motion data for standard formats used in the TRAF system. The system is able to recognise models of motion captures in Acclaim, Biovision and Hanim formats. After normalization data is analysed to check for content of elementary motions. Two types of motions are considered: uniform and cyclic. If a velocity of a given joint movement is zeroing in at least one of the three dimensions then this type of motion is classified as a uniform motion. Applied algorithms are based on a process of joints' displacement and angles between bones analysis. For detection of cyclic motions a method called detection of full motion cycles [14] extended by calculations of weighted fitness error.

3.2 Represenatations of Movement

A human figure is represented in the computer memory store in a form of exosceletal hierachical structure. This means that to describe an animation made of n frames we need to record for each of them locations of all m bones of the figure in 3D space [15,16]. Following this method m-dimentional data representing a given time sequence of translations and hierarchical rotations is created.

For the model to be useful a distance metrics need to be defined. A simplified metrics method was proposed [14] that was used for parametric models definition to allow modification of a motion's specific character. The SM model relies on definition of the animated figure's motion using the curve gluing method [5]. The chain of rotation chain for a given skeletal joint is a sequence of data that changes in discrete point in time domain. Using SM approaches both interior and exterior parameters were defined for each of the motion sequences.

An input representation is original motions capture sequence – which is represented by Raw Data Model. This model uses Euler's angles as data representation, which is highly inconvenient for further computations. From RDM we extract shorter Primitive Motions (PM), which are uniform by means of the similarity of motion character. We also propose a set of different representations, which are more suitable for this kind of data. Data using one of those representations forms a Primitive Instance (PI). In Absolute Translations (AT) representation, each joint position is stored in each frame as translations that are absolute in regard of global coordinates. To overcome the restrictions of AT representation, data can be saved in a relative way. It leads to Relative Translations (RT) representation. We have noticed that motion capture data can be represented in a very flexible way using quaternions. According to [20] we use them to represent rotation in three-dimensional space:

$$q = w + x_i + y_j + z_k = [w\ x\ y\ z] = (s,\ v) \tag{1}$$

Quaternions are naturally represented in 4-dimensional complex numbers. Their characteristics in motion capture domain are described in details in [13].

A gimbals'-lock effect is eliminated and computational errors do no cumulate as in standard representation of rotation matrices. Thus a motion representation is called Quaternion Based Model (QBM). A motion sequence is represented as the following matrix:

$$v_{\max} = \begin{bmatrix} q_{1,1} & q_{1,2} & \cdots & q_{1,n} \\ q_{2,1} & q_{2,2} & \cdots & q_{2,n} \\ \vdots & \vdots & q_{i,j} & \vdots \\ q_{m,1} & q_{m,2} & \cdots & q_{m,n} \end{bmatrix} \tag{2}$$

where m denotes number of bones and n is number of frames in motion sequence. In QBM each element q i. j of the matrix V corresponds to transformation of the bone indexed as i in frame number j.

A clustering algorithm uses the Primitive Instance, which includes data converted into one of the above representations, in order to obtain some groups of motions. Next, a probabilistic Generic Model (GM) is evaluated for each motion group. GM is a set of parameters described by Gaussian distributions over parameters of Primitive Instances of a particular group. From these distributions new PM's can be generated. Fig.1 shows a process of determining various motion models. More detailed description of each model is described in [1].

3.3 The Comparison of Movements

In order to find two compatible elementary motions in the process of motion, we needed to design a suitable method for comparing two movements. In the paper [17] authors defined the description metrics such as: (a) similarity of singular motion frames, (b) Similarity of specific elementary motions that originate from comparisons of their internal character(s).

Thanks to the above metrics various approches were applied to define methods of comparing individual motion frames as well as task of marking of similarities in terms of their internal character. The comparison gets more complicated when motion sequences have varying length. Instead of scaling, for data representing the motion an algorithm of Dynamic Time Warping (DTW) can be applied. The DTW algorithm relies on dynamic programming technique where a test sequence is warped in relation to the original sequence pattern. A good description of the DTW algorithm can be found in [17] and examples of its various adaptations for data comparisons in motion capture in [14]. Different kind of metrics can be used to compare two isolated points (classical, derivative based as in the works [19]). However, when applying Derivate method for data that describes the movements such metrics causes errors in the order of both motion sequences. It's worth to note that interesting results can be obtained when both the signal's value (the classic DTW algorithm) and derivative of the signal (DDTW algorithm) propositions are considered. The extended adaptation of the algorithm is called VDDTW (ang. Value Derivative Dynamic Time Warping).

3.4 User Interaction

The general aims for the project was to: (1) be to enable the users to define various motion related tasks for the TRAF system, (2) enable a set-up of limits/constraints that characterise the definition of a given task, (3) presenation of obtained results.

The user definition of a given task include marking of so called control points decribed by combination of time, place and posture. A figure's posture at a control point constitutes a real frame of a given motion animation. Between control points we define behavioual segments (blocks). The user can define behavioual segments her/himself. A given behavior needs to relate to groups of similar character that is defined by DB module. The IU Module constructs a set of restrictions based on the control test data and behavioual segments.

Obstacles can be placed on the stage hence we need to consider to deifne moevemnt trajectory of the figures [12] between the control points. For this purpose global path planning methods widely used in robotics can be adapted. After processing in the Sythesis Module of motion data is available for a display in a dedicated graphic envrionment.

4 Problems Related to Synthesis of Motion

The task of synthesizing the motion relies on attempt to recreate a motion using derived collection of limits. The final result is generated thanks to exchange and information processing between User Interaction and Data Base Modules. The process of finding a solution consists of the following several stages:

1. Analysis of limitations – it divides the collection of limits into unit tasks.
2. Multilevel approach – allows to view motion data on various levels of abstraction.
3. Finding simple solutions – this helps to resolve the user problem by splitting it into several atomic tasks.

To obtain an accurate solution we need to perform an analysis of limits and on this basis define further steps of the algorithm. These limits are transmitted if

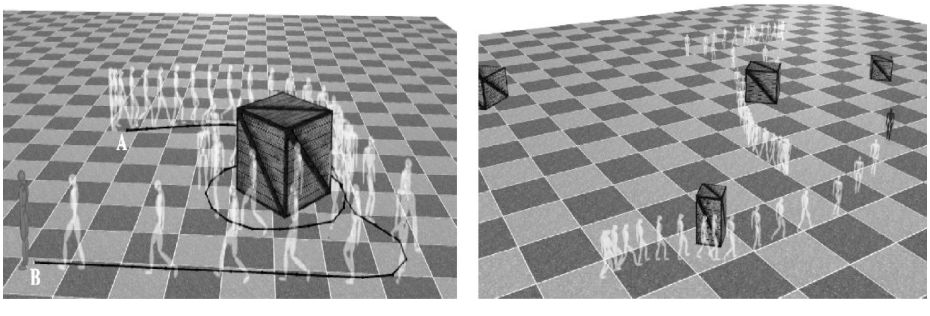


Fig. 3. Walk around the obstacles Generation of complex motion [13]

form of enquiries about motion models to DB Module. Thanks to this approach we can automate the process of finding the solution as well as and derive a partial solution in cases when it isn't possible to find a satisfactory solution for all elementary solution. Examples of mechanisms typically used in generation of motions in our experiments are copies of screen images taken in ARENA system (a newer version of TRAF) here shown in Figures 3 (courtesy of the original author [13]).

5 Conclusion

In this paper was introduced a new toolkit for automatic generation movement animation. The movement animation is a relatively new discipline hence there is a shortage of useful software toolkits and proven animation techniques and applicable standards. Over the project lifetime we were able to resolve many difficult technical problems and bring forward several innovative concepts and design solutions. So far not all of them found practical applications. At the present stage of the development of the TRAF system we specifically highlight the need to provide optimal solutions for automatic generation of human figure motion with the focus on aspects motion realism that is to be derived from the obtained motion sequences by reducing amount of constraints set on the animation by the user. Brand and Herzman [4] described similar system in properties to the ideas of TRAF They were looking for patterns in a motion set and afterwards use them to generate any animation. However specific solutions are different. Brand and Herzman take advantage of a probabilistic motion model. In the TRAF system, a statistical description, which is computed for each group of similar motions, is designed only for motions' comparison. The synthesis itself is based on original sequences. Extensive empirical testing will be required to determine which of these two systems gives better results.

References

1. Amaya, K., Bruderlin, A., Calvert, T.: Emotion from Motion. In: Craphics Interface 1996, pp. 222–229 (1996)
2. Arikan, O., Forsyth, D.A.: Interactive motion generation from examples. In: Proc. of ACM SIGGRAPH (2002)
3. Bak, A.: Organizacja bazy danych ruchow elementarnych animowanych postaci udzkich, Praca Magisterska, M.Sc.Eng. Thesis (in Polish), Wroclaw University of Technology (2001)
4. Brand, M., Hertzmann, A.: Style machines. In: Proc. of ACM SIGGRAPH (2000)
5. Cohen, M., Bodenheimer, B., Rose, C.: Verbs and Adverbs: Multidimensional Motion Interpolation. IEEE Computer Graphics and Applications 18(5), 32–40 (1998)
6. Burgielski, Z.: Interakcja z użytkownikiem w systemie analizy i syntezy ruch animowanych postaci ludzkich, M.Sc.Eng. Thesis (in Polish), Wroclaw University of Technology, Faculty of Electronics (2001)
7. Chaczko, Z.: NIE Models for Emergency Services Interoperability. In: Proceedings of Computer Aided Systems Theory: EuroCAST 2007. LNCS, Springer, Heidelberg (to be published, 2007)

8. Chaczko, Z., Sinha, S.: Strategies of Teaching Software Analysis and Design – Interactive Digital Television Games, ITHET 2006, Sydney, Australia (July 2006)
9. Cutler, R., Davis, L.: Robust Real-Time Periodic Motion Detection, Analysis, and Applications. IEEE Transactions on Pattern Analysis and Machine Intelligence (2000)
10. Dyer, S., Martin, J., Zulauf, J.: Motion Capture White Paper (12 December 1995), http://reality.sgi.com/jam_sb/mocap/MoCapWP_v2.0.html-HDR0
11. Gleicher, M.: Motion editing with space-time constraints. In: Symposium on Interactive 3D Graphics, pp. 139–148 (1997)
12. Isaacs, P.M., Cohen, M.F.: Controlling dynamic simulation with kinematics constraints, behavior functions and inverse dynamics. Computer Graphics 21(4), 215–224 (1987)
13. Jabłoński B.: Metody porøwnywania generator ruchu animowanych postaci ludzkich, M.Sc.Eng. Thesis (in Polish), Wroclaw University of Technology, Faculty of Electronics (2003)
14. Jabłoński, B.: Comparison of human motion generators. In: Klempous, R. (ed.) Diversitas Cybernetica, Warsaw. Lecture Notes in Communication series, pp. 101–119 (2005)
15. Jabłoński B., Klempous R., Majchrzak D.: Models and methods for biometric motion identification, Annales UMCS, vol. 4, Lublin (2006)
16. Jabłoński, B., Klempous, R., Majchrzak, D.: Feasibility analysis of human motion identification using motion capture. In: Proc. of the Int. Conference on Modelling, Identification and Control, Acta Press (2006)
17. Jabłoński, B., Kulbacki, M., Klempous, R., Segen, J.: Methods for comparison of animated motion generators. In: Proc. of IEEE International Conference on Computational Cybernetics, W egry (2003)
18. Keogh, E.J, Pazzani, M.J.: Derivative Dynamic Time Warping. In: First SIAM International Conference on Data Mining, Chicago (2001)
19. Li, Y., Wang, T., Shum, H.Y.: Motion texture: a two-level statistical model for character motion synthesis. In: Proc. of ACM SIGGRAPH (2002)
20. Maillot, P.: Using quaternions for coding 3d transformations. In: Graphics Gems I, pp. 498–515. Academic Press Inc, Boston (1990)
21. Pullen, K., Bregler, C.: Motion capture assisted animation: texturing and synthesis. In: Proc. of ACM SIGGRAPH (2002)

An Annotation Tool for Video Understanding

M. Rincón and J. Martínez-Cantos

Dpto. de Inteligencia Artificial. ETSI Informatica. UNED.
Juan del Rosal 16, 28040 Madrid, Spain
mrincon@dia.uned.es, javiermcantos@bec.uned.es

Abstract. The interest for developing annotation tools for interpretation of video sequences arises on the own necessity of conceptualizing scenes with a suitable degree of semantics for the application domain. In this paper we analyze the features that must be present in a video annotation tool for video understanding: the entities of interest that appear in each description level of the scene are analyzed and primary features (those that must be annotated initially) and derived features (those obtained automatically) are distinguished. Lastly, we present a video annotation tool based on the previous analysis, for which the design we have chosen is modular, reusable and user friendly.

Keywords: annotation tools, video understanding, ontology based model definition, semantic description, computer vision.

1 Introduction

One of the reasons for the development of tools for video annotation is the necessity of conceptualizing images with the suitable degree of semantics for the application domain [1]. Nowadays, one of the biggest challenges in artificial vision and video understanding is the description of scenes where many agents with complex dynamics play and interact each other, e. g. humans or their parts, in applications such as surveillance, forensic analysis, intelligent human-computer interfaces, etc.

Since the behaviors of these agents are complex, the related models are difficult to fit analytically, which forces to incorporate a large amount of domain knowledge and the use of approaches based on case-based learning. The annotation tools allow us to enrich the description of specific scenarios adding domain information, as well as to define case bases for subtasks that requiere learning or for system evaluation.

In this paper, a video annotation tool for video understanding is described. This has been created inside the AVISA project [2], which aim is to generate a surveillance task oriented description of video sequences from several cameras deployed over a scenario. Hence, an architecture of description levels with increasing degree of semantics is used to split the interpretation problem [3][4], making easier the representation of the entities and the inferences inside each level.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 701–708, 2007.
© Springer-Verlag Berlin Heidelberg 2007

This paper is organized as follows. In section 2, the different description levels involved in the task oriented description are analyzed in order to obtain those features that are interesting to annotate. Next, in section 3, the proposed tool -oriented to usability and reutilization - is described. The generic knowledge of the application has been separeted from the specific knowledge of the domain by means of metamodels that are particularized easily for each description level and application domain. In section 4, there are examples and comparatives with other well-known image annotation tools. Finally, section 5 analyses the results obtained and the future work proposed for improving the proposal.

2 Video Understanding Analysis

As mentioned above, in order to reduce the semantic gap between the input video signals and the surveillance task-oriented scene description, different description levels with increasing degree of semantics have been used in the AVISA project, facilitating the inferences and allowing to pass from one level to the next until the scene description in the appropriate level is reached, i. e. the task level. The involved description levels are: the image level, the blob level, the object level and the activity level. In this section, the features of the entities belonging to each level are analyzed including the motivation for their annotation in training sequences or in the definition of scenarios for video understanding.

We will distinguish two types of features: the primary ones and those derived. On one hand, the primary features must be declared in an explicit manner, e. g. the position or the instant of time, the movement capability of an object, etc. On the other hand, the derived features, e. g. texture, average color, convex hull, etc., can be obtained automatically from the primary ones. Therefore, our study is focused over the primary features, i. e. those that it will be necessary to annotate in a manual or semiautomatic manner.

At the image level, the features of specific pixels are represented and, though it is necessary to keep it in mind in the design of a complete system, that information is closely related to the physical sensor and it doesn't contain outstanding information for the task oriented description that should be annotated.

At the blob level, the groups of pixels that share certain properties are represented, i. e. regions and lines. These entities are associated with features of the interesting objects that are visually perceptibles (position, color, texture, size, etc.) or features of the scenario (region of interest, region of exclusion, etc.). Annotating these regions allow to simplify, configure and evaluate tracking and object recognition algorithms in complex situations, e. g. treatment of groups, occlusions, changes of illumination, etc. As primary features of the regions, we consider their position in each time instant. The position can be represented by means of a representative point of the region (interior point, center of masses, invariant point, bounding box, etc.) or, in a more precise way, with their contour (list of points). The annotation of regions will depend on the features of the region and the annotation objective. For example, if we want to indicate the position, a representative point or the bounding box would be enough; if the

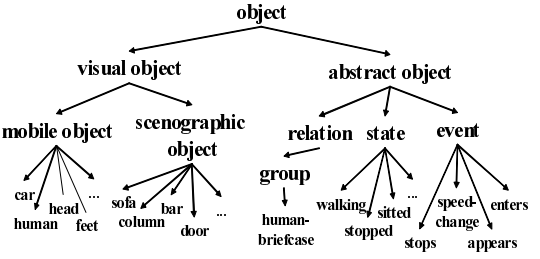


Fig. 1. Classification of entities managed at the object level

region presented uniform features, the indication of an inner point could be used, because it is possible to obtain the related region automatically; but if a precise definition of the region is required, and the previous uniformity conditions are not achieved, it will be necessary to trace its contour.

Once the regions of interest of the image have been obtained, these are associated with the interesting objects present in the scene at the object level. Thereby, we get a description of the scene called GSD [5], which includes the representation of the objects present in the scene and the visual, geometric and spatial relationships among them (Figure 1).

In order to annotate the sequence, it will be interesting to declare, for each object, the dynamic features of the whole object or some of its components (position in each instant of time, if it is visible in the image, if it occludes other objects, its movement type, etc.), as well as the static properties of the objects that restrict their variability (capacity to occlude, movement capacity, capacity of shape transformation, capacity of color change, etc.). These features are used to improve the algorithms of recognition and tracking of specific objects, because they allow the feedback from the object level to the blob level [6].

With regard to the relationships among objects, these are modeled as abstract objects. These are from the grouping of visual objects that play specific roles in the relationship. For example, within a group of humans staying inside a car (abstract-entity = "Vehicle-passengers"), different roles ("driver", "co-driver", etc.) can be distinguished.

A special type of relationship is that one stablished between an object and its parts. It is interesting to organize this composition relationship inside a model of the object, declaring attributes like completeness, exhaustiveness, disjoint, structure, etc. This explicit declaration of the model facilitates the annotation of the present objects in the scene, e. g. to delimit adjacent regions by means of a single contour, to maintain a control of the objects that is necessary to annotate, etc.

Another type of relationships is the quantitative geometric relationship that can be obtained automatically from simple operations over the present instances in the scene, e. g. the distance between two objects, the object trajectory, the relative position of two objects, etc. Therefore, these relationships are derived and it is not necessary to annotate them initially.

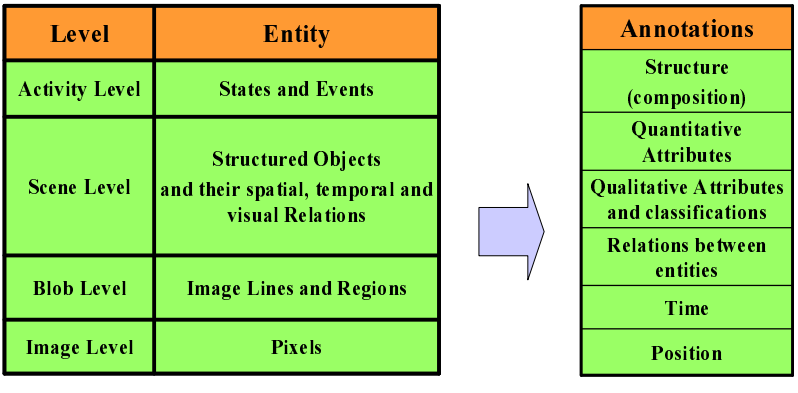
Regarding the objective of simplifying the description of the GSD and facilitating the control of the information, it can be carried out a more abstract description of the scene by means of states and events that describe specific configurations of spatio-temporal features (e. g. the human X "is walking", "has stopped", etc.). This allows to condense the spatio-temporal information in order to simplify the description and to filter certain "noise" that appears when monitoring the low level variables due to the use of simplified descriptions carried out in current artificial vision methods, such as fixed and reduced temporal windows, few significant variables, simple rules, etc.

Some of these states and events transcend to the superior level as primitive building blocks for the task-oriented scene description. At the activity level, the states and events of the present objects in the scene are composed to describe more and more complex activities until the description level pursued by the task is achieved. The managed entities at this level are abstractions of the spatio-temporal features of the objects present in the scene or the relationships among them.

On one hand, just as it is used in [7], the term "event" groups both the specific events and the states. The states are annotated by means of the initial and final instant declaration, while the specific events are annotated with a single instant of time. The events can have quantitative or qualitative attributes and they are associated to the visual objects that are present in the scene or to abstract objects that define relationships. For example, the state "walking" can be associated to a certain human X between two instants of time, ts and tf, and also it can have quantitative (e. g. the speed) or qualitative attributes (e. g. a classification of the speed in {slow, normal, quick}).

On the other hand, there can be simple and composite events. An event is simple if it is obtained directly from spatio-temporal features of the objects described in the previous level. An event is composed when it is described as a composition of simpler events (i. e. simple or composite events). When annotating, it occurs the same situation than at the object level: the composite event

Table 1. Annotations of the entities of the description levels for video interpretation



description presents properties as completeness, sequenciality and overlapping, and the class-subclass relationship presents properties as exclusivity and exhaustivity. The explicit declaration of these features facilitates the annotation of the scene, because they must be declared only one time. For example, if a composition relationship exists in which an event Ex is described by means of other events {Ex1,Ex2,Ex3} with a sequential, continuous and without overlapping organization, then, it is possible to use the knowledge about that event Ex, to annotate the component events, because the end of one event coincides with the beginning of the next one (e. g. tEx2ini = tEx1end).

Table 1 summarizes the annotations of interest for video interpretation of the managed entities in the different description levels.

3 The Annotation Tool

In the design of the video annotation tool, a modular solution has been chosen taking into account that it must be reusable for different video understanding tasks and user friendly (Fig. 2). The tool is designed for manual annotation of the scenes, but it also provides a simple interface to import partial annotations from external tools. So, it is possible to test new functionalities before integrating them in the annotation tool.

Also, from the explicit annotation of the primary features, it is possible to define automatically other qualitative or quantitative features that are of interest for a certain application.

This annotation tool can be used for multiple purposes inside the video understanding process, e. g. groundtruth definition for evaluation of algorithms, generation of case bases for learning-based object or situation modelling, task-orientated object-level background description (such as scenographic objetcs in context), etc.

3.1 Internal Structure

According to the analysis carried out in the section 2, it is deduced that there are generic properties of the entities of interest in each description level that are convenient to declare in a previous model, because they facilitate reutilization, documentation, analysis and annotation of the video sequence.

Therefore, the annotation knowledge has been separated from the domain knowledge. This information is gathered in a metamodel that contains the descriptions of the types of objects that can be used to describe the instances involved in a specific scene. The metamodel has been initially defined in XML, although we keep in mind to rewrite it using OWL.

3.2 Evaluation

An example of use of the tool is shown in figure 3. The interface of the annotation tool is used for annotating an image where different types of objects are depicted.

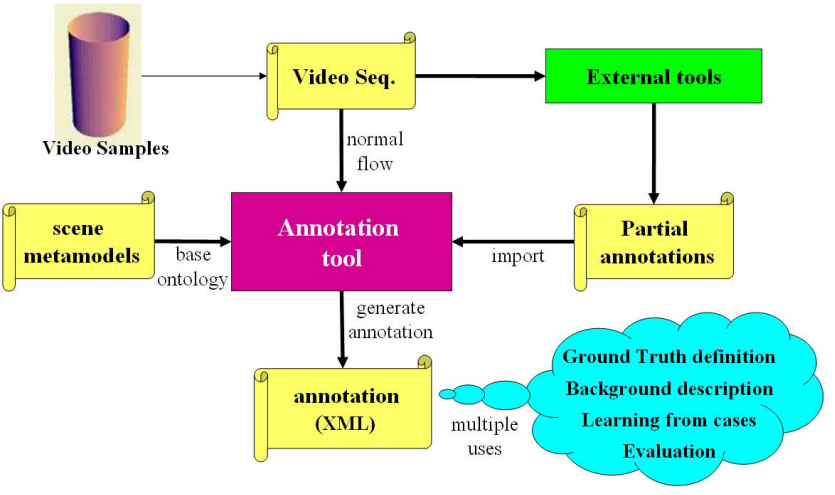
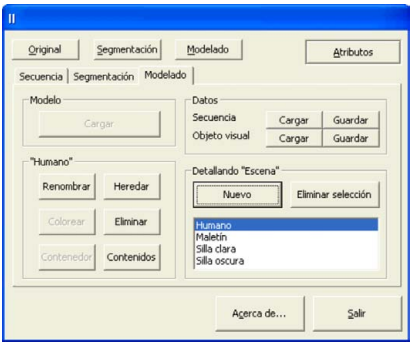
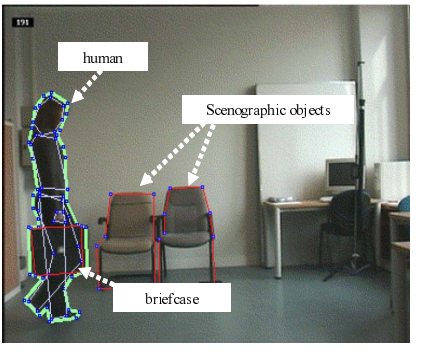


Fig. 2. Annotation tool architecture



(a)



(b)

```
<Scene type="visual object">
  <Human type="visual object" min="0" max="100">
    <Animal type="visual object" min="0" max="100">
    <Vehicle type="visual object" min="0" max="100">
    <Scenographic_element type="visual object" min="0"
      max="100">
</Scene>

<Human type="visual object">
    <Head type="visual object" min="0" max="1">
    ...
    <Leg type="visual object" min="0" max="2">
</Human>

<Head type="visual object">
    <Eye type="visual object" min="0" max="2">
    <Nose type="visual object" min="0" max="1">
    <Mouth type="visual object" min="0" max="1">
    <Ear type="visual object" min="0" max="2">
</Head>
```

(c)

```
<Scene name="Human with Briefcase">
  <Human name="Marga">
    <PositionInstant name="miniJPG023.bmp">
      <Contour>(10,260)(7,268)(3,273)(2,262)(10,260) ...</Contour>
    </PositionInstant >
    ...
    <PositionInstant  name="miniJPG026.bmp">
      <Contour>(1,271)(13,273)(14,277)(0,283)(1,271) ... </Contour>
    </PositionInstant >
    <PositionInstant  name="miniJPG027.bmp">
      <Contour>(1,72)(12,86)(8,89)(3,87)(3,120)(7,128)(6,174)
               (26,236)(19,238)(13,239)(13,249)(17,265)(28,268)
               (13,276)(11,280)(3,283)(0,287)(1,72)</Contour>
    </PositionInstant >
    <PositionInstant  name="miniJPG028.bmp">
      ...
```

(d)

Fig. 3. Annotation tool example: a) annotation tool interface; b)modeled instances in a scene; c) human model descriptcion; d) annotation instance example

Table 2. Comparison of video annotation tools for video understanding

	ELAN [8]	Video-AnnEx [9]	CAVIAR [10]	ViPER-GT [11]	AVISA
Audio support	x				
Video support	x	x	x	x	x
Time-stamped annotations	x	x	x	x	x
Spatial annotations		Rectangular region	Bounding box circles	Single contour	Set of contours
Use of model information			x	x	x
Other facilities	independent and referring events, analysis of annotations (search)	Annotation of events, static scene and key objects	Group support, copy features to next frame, consistency check	Propagation, interpolation, undo	Inheritance, segmentation, import partial annotations
Metadata	XML	MPEG-7 over XML	XML	XML	XML

One of these objects is a "human", so we use a human model to annotate this specific instance at different position instants.

Nowadays, we try to keep the system open and flexible and, in spite of increasing the annotation time, we are not considering the features of the composition relationship among objects for the annotation (they are only used for information organization).

In table 2, we compare our video annotation tool with other similar tools. There are tools for the treatment of the image and their decomposition in regions of interest, but these are not user friendly when treating video sequences, other tools are designed for event annotation (time-stamp annotations), but not considering the special features related to events, such as spatial location. Other tools pursue different objectives, for example, the CAVIAR project provides a tool for the annotation of video sequences, but in a simple way, because it is more oriented to comparative studies than to system tuning by learning. Others focus on MPEG-7 annotations for indexation of multimedia files.

4 Conclusions

This paper has analyzed the features that a video annotation tool oriented to video understanding must present. Initially, in order to generate case data bases for model fitting of complex activities, we have distinguished the primary features of the entities of the different description levels, which are necessary to annotate manually or semiautomatically. Regarding that, it is possible to carry out annotations of derived features automatically. The proposed tool structure has been described and, finally, it has been compared with other annotation tools.

In this first prototype of the tool, reusability has prevailed over usability. However, the tool design is modular and, we will improve the mentioned factors in future versions as we will improve the metamodel.

Acknowledgments

The authors would like to thank the CiCYT for financial support via project TIN-2004-07661-C0201 and the UNED project call 2006.

References

1. Buxton, H., Mukerjee, A.: Conceptualizing images. Image and vision computing 18 (2000)
2. Mira Mira, J.: Memoria del proyecto TIN2004-07661-C0201, AVISA: Diseño e implementación de un conjunto de agentes de diagn óstico, planificación y control, con capacidad de aprendizaje y cooperación con humanos en tareas de vigilancia. UNED (2004)
3. Nagel, H.H.: Steps towards a cognitive vision system. AI Magazine 25(2), 31–50 (2004)
4. Bachiller, M., Martínez, R., Mira, J., Rincón, M.: Niveles de Descripción para la Interpretación de Secuencias de V ídeo en Tareas en Vigilancia. CMPI'2006 (Campus Multidisciplinar en Percepción e Inteligencia) (2006)
5. Fernyhough, J., Cohn, A.G., Hogg, D.C.: Constructing qualitative event models automatically from video input. Image and Vision Computing 18, 81–103 (2000)
6. Rincón, M., Carmona, E.J., Bachiller, M., Folgado, E.: Segmentation of moving objects with information feedback between description levels. In: IWINAC 2007, Part II. LNCS, vol. 4528, pp. 172–181 (2007)
7. Bolles, B., Nevatia, R.: A hierarchical video event ontology in OWL. Final report, ARDA Project (2004), `https://rrc.mitre.org/nwrrc/OWL-events-final-report.pdf`
8. ELAN annotation tool: `http://www.lat-mpi.eu/tools/elan/`
9. VideoAnnEx annotation tool: `http://www.alphaworks.ibm.com/tech/videoannex`
10. CAVIAR Project: `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`
11. VIPER-GT annotation tool: `http://viper-toolkit.sourceforge.net/`

Temporal Constraints in Large Optical Flow Estimation

Agustín Salgado and Javier Sánchez

Computer Science Department,
University of Las Palmas de G.C.
35017 Las Palmas de G.C., Spain
{asalgado,jsanchez}@dis.ulpgc.es
http://serdis.dis.ulpgc.es/~jsanchez

Abstract. The aim of this work is to propose a model for computing the optical flow in a sequence of images. We introduce a new temporal regularizer that is suitable for large displacements. We propose to decouple the spatial and temporal regularizations to avoid an incongruous formulation. For the spatial regularization we use the Nagel–Enkelmann operator and a newly designed temporal regularization. Our model is based on an energy functional that yields a partial differential equation (PDE). This PDE is embedded into a multipyramidal strategy to recover large displacements. A gradient descent technique is applied at each scale to reach the minimum.

1 Introduction

In this paper we consider the problem of estimating the optical flow assuming that the objects may undergo large displacements. The difference with respect to other related approaches is that we exploit the temporal dimension of the sequence. Most of the better known methods only deal with the problem of estimating the optical flow between two frames, ignoring that the sequence comprises several images which are all related. We show in this paper that it is possible to integrate the temporal information to improve the results.

We propose a variational technique in where an energy functional is minimized yielding a diffusion–reaction PDE. These kind of energy–based approaches have been largely used in optical flow estimation. Horn and Schunck [8], for instance, propose to minimize the so–called optical flow constraint equation (OFC) together with a smoothing term depending on the optical flow gradient. Later some authors have proposed several improvements to overcome the shortcomings of this method like in [3], [6], [11], [7].

In order to compute large displacements a common strategy is to use a multi–pyramidal decomposition like in [3] in where each scale is represented by decreasing size of images.

The first works on spatio–temporal methods are due to Nagel [10] and Black and Anandan [4]. More recently in [13] and [5] the authors propose similar continuous models for the spatio–temporal smoothness constraint.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 709–716, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In this work we propose a novel temporal regularizing term which is explicitly designed to support for large displacements. The previous mentioned spatio–temporal methods treat the spatial and temporal dimensions in the same way. We show here that when large displacements are present, it is not suitable to use temporal derivatives. It is more convenient to separate the temporal smoothing term. We also show that the major contributions of spatio–temporal smoothness are given by more stable and accurate results. This is clearly stated in the experimental results in where some comparisons with its corresponding spatial method is carried out on synthetic and real sequences.

In Sect. 2 we give an overview on related optical flow methods and a study on the generalization of spatial optical flows and justify the use of temporal regularizers. In Sect. 3 we derive the numerical scheme for the energy proposed in the previous section and in Sect. 4 we demonstrate the performance of our method by using synthetic and real sequences and compare with spatial methods. Finally in Sect. 5 the conclusions.

2 The Method

The optical flow, $\mathbf{h}(\mathbf{x}) = (u(\mathbf{x}), v(\mathbf{x}))^T$, is the apparent motion of pixels in a sequence of images. One of the first in introducing a variational formulation for the computation of the optical flow was Horn and Schunck [8]. They proposed the so–called optical flow constraint equation $\frac{dI(\mathbf{x},t)}{dt} = \nabla I \cdot \mathbf{h} + I_t = 0$, which states that the image intensity remains constant through the sequence –known as the Lambertian assumption–.

This equation is valid when the object displacements in the scene are continuous. This is not the case in our approach since the objects may undergo large displacements. A different formulation which is the corresponding to the optical flow constraint equation in the discontinuous case is $I_1(\mathbf{x}) - I_2(\mathbf{x} + \mathbf{h}(\mathbf{x})) = 0$. Other approaches different from the quadratic form are commonly used (see for instance [9] and [12]).

Typically, the data term is accompanied by a smoothness term depending on the gradient of the flow. Horn and Schunck [8], for instance, proposed to minimize the square norm of the optical flow gradient $\|\nabla \mathbf{h}\|^2 = \|\nabla u\|^2 + \|\nabla v\|^2$ which provides smooth solutions and yields an isotropic diffusion equation at the PDE. This model has been improved during the last years and some authors has introduced some different approaches in order to respect the image or flow discontinuities. A well–known approach is that of Nagel–Enkelmann [11] that introduce an operator depending on the image gradient that enables anisotropic diffusion in order to respect the object contours. Other improvements are given by the methods explained in [2], [6] and [7].

Variational temporal regularizers have been introduced in some works like in [4] and [10]. More recently Weickert and Schnörr [13] introduced a method in where they used the continuous optical flow constraint equation as data term and a regularizing term of the form $\mathcal{R}(|\nabla_3 u|^2 + |\nabla_3 v|^2)$. In this case the temporal derivative is treated in the same manner as the spatial derivatives.

We separate the temporal and spatial derivatives. In the spatial domain we use the Nagel–Enkelmann operator and for the temporal domain the functional $T(\mathbf{h}_i, \mathbf{h}_{i+1}) = \Phi(\|\mathbf{h}_i - \mathbf{h}_{i+1}(\mathbf{x}+\mathbf{h}_i)\|^2) + \Phi(\left\|\mathbf{h}_i - \mathbf{h}_{i-1}(\mathbf{x}+\mathbf{h}^*_{i-1})\right\|^2)$ including the backward optical flows, $\mathbf{h}^*$ as

$$\begin{aligned} E(\mathbf{h}) = & \sum_{i=1}^{N-1} \int_{\Omega} \left(I_i - I_{i+1}\left(\mathbf{x}+\mathbf{h}_i\right)\right)^2 \, d\omega \\ & + \alpha \sum_{i=1}^{N-1} \int_{\Omega} trace(\nabla \mathbf{h}_i^T \mathbf{D}(\nabla I_i) \nabla \mathbf{h}_i^T) \, d\omega \\ & + \beta \sum_{i=1}^{N-2} \int_{\Omega} \Phi(\|\mathbf{h}_i - \mathbf{h}_{i+1}(\mathbf{x}+\mathbf{h}_i)\|^2) \, d\omega \\ & + \beta \sum_{i=2}^{N-1} \int_{\Omega} \Phi(\left\|\mathbf{h}_i - \mathbf{h}_{i-1}(\mathbf{x}+\mathbf{h}^*_{i-1})\right\|^2) \, d\omega \end{aligned} \tag{1}$$

where $\Phi(x^2) = 1 - \gamma e^{-\frac{x^2}{\gamma}}$. When displacements are very small, $\mathbf{h}_i - \mathbf{h}_{i+1}(\mathbf{x}+\mathbf{h}_i)$ is an approximation of the temporal derivative.

With the last two integrals we are implicitly assuming a model for the object velocities. We enforce that the $\mathbf{h}_i$ functions be similar in magnitude and direction, so this scheme is more suitable for objects that move with constant velocity in a permanent direction.

3 Minimizing the Energy

A functional variation of our energy leads to the Euler–Lagrange equations:

$$\begin{aligned} \mathbf{0} = & -\left(I_i(\mathbf{x}) - I_{i+1}(\mathbf{x}+\mathbf{h}_i)\right) \nabla I_{i+1}(\mathbf{x}+\mathbf{h}_i) \\ & -\alpha \left(\mathbf{div}(\mathbf{D}\left(\nabla \mathbf{h}_i\right) \nabla u_i), \mathbf{div}(\mathbf{D}\left(\nabla \mathbf{h}_i\right) \nabla v_i)\right)^T \\ & +\beta \, \Phi'(\|\mathbf{h}_i - \mathbf{h}_{i+1}(\mathbf{x}+\mathbf{h}_i)\|^2) \\ & \cdot \Big(\left(\mathbf{h}_i - \mathbf{h}_{i+1}(\mathbf{x}+\mathbf{h}_i)\right)^T \left(\mathbf{Id} - \nabla \mathbf{h}^T_{i+1}(\mathbf{x}+\mathbf{h}_i)\right) \Big) \\ & +\beta \Phi'(\left\|\mathbf{h}_i - \mathbf{h}_{i-1}(\mathbf{x}+\mathbf{h}^*_{i-1})\right\|^2) \\ & \cdot \left(\mathbf{h}_i - \mathbf{h}_{i-1}(\mathbf{x}+\mathbf{h}^*_{i-1})\right) \end{aligned} \tag{2}$$

We apply a gradient descent technique to reach the solution of the previous system of equations and embed it into a multi–pyramidal approach to deal with large displacements. We create a pyramid of scales for the whole sequence with different size of images and solve the previous system of equations at each level. Once we obtain a stable solution for a scale we use this as a first estimate for a finer scale.

Therefore we have a number of scales $s_1, s_2, \ldots s_n$. At each scale we solve the previous system of equations for the whole set of unknows $\{u_i^s, v_i^s\}$ and then we use this as a first approximation for the following scale $\{u_i^{s_1}, v_i^{s_1}\} \rightarrow \{u_i^{s_2}, v_i^{s_2}\} \rightarrow \ldots \{u_i^{s_n}, v_i^{s_n}\}$.

4 Experimental Results

The purpose of these experimental results is to show the performance of our novel method with respect to its spatial counterpart. We are also interested in comparing this new method with its simple form given by the forward temporal cost matching function $\mathcal{T}(x) = \Phi(\|\mathbf{h}_i - \mathbf{h}_{i+1}(+\mathbf{h}_i)\|^2)$. In the sequel we name the spatial method as "Spatial", the simpler temporal method as "Temporal" and the complete temporal method as "Bi–Temporal" or "Bidirectional–Temporal" – Eq. (1) –.

4.1 Translating Square

The first sequence consists of ten frames with a black square moving ten pixels horizontally forward with a constant velocity over a white background. In Fig. 1 we can see several frames of the sequence and in Fig. 2 we show the angular error for all the frames using the three methods. In table 1 we show the mean angular and euclidean errors.

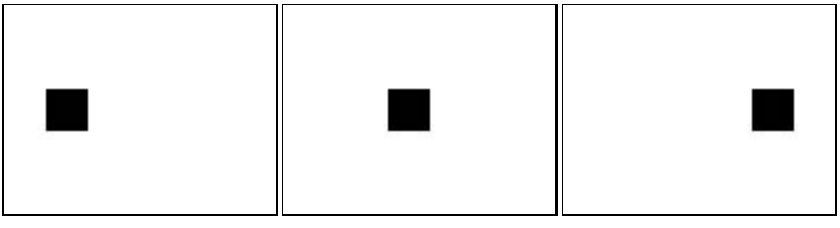


Fig. 1. Frames 0, 4 and 9 of the square sequence

From table 1 and Fig. 2 we can see that the temporal methods are very stable: the euclidean and angular errors for all the frames are very similar. However, the "Spatial" method is clearly unstable.

The improvement of the "Temporal" method with respect to its spatial counterpart is about 13,79% of the angular error and in the case of the "Bi–Temporal" is about 24,14%. The euclidean error for the "Temporal" method is bigger than the spatial method. This is justifiable because the last optical flow has a big error with respect to the rest of frames. The simple temporal method does not improve the last solution because it does not receive information from other frames. Even more, when this last estimate is bad, it propagates the bad guests to the rest of sequence. In the case of the "Bi–Temporal" method the improvement with respect to the spatial one is about 31,58%. This method is not affected by the last optical flow as in the previous case.

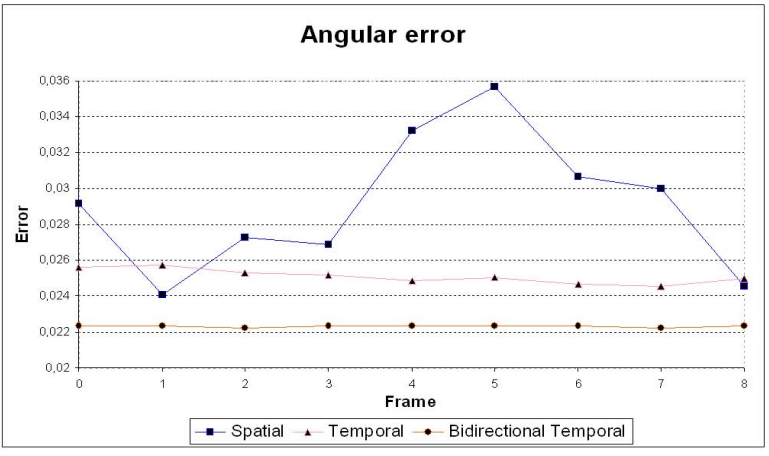


Fig. 2. Angular error for the square sequence

Table 1. Mean angular and euclidean errors for the square sequence: AE_μ is the mean angular error and AE_σ its standard deviation; EE_μ is the mean euclidean error and EE_σ its standard deviation

Method	AE_μ	AE_σ	EE_μ	EE_σ
Spatial	$0{,}029^o$	$3{,}6E\text{-}3^o$	0,0189	0,013
Temporal	$0{,}025^o$	$3{,}7E\text{-}4^o$	0,0246	3,8E-4
Bi-Temporal	$0{,}022^o$	$3{,}0E\text{-}5^o$	0,0132	1,9E-4

4.2 Marble Blocks

We used the Marble Block sequence – Fig. 3 – for the second test.

In Fig. 4 we can see the ground truth and the flows obtained with the spatial and temporal methods.

If we look at this image we observe that both solutions are similar but there are some differences: The floor is smoother in the temporal disparity map and the background is more respected. The right tower that disappears seems more continuous but, on the other hand, in the middle of the two left–most towers the disparity maps seem under–estimated. Globally, the temporal method seems to have better estimated the object discontinuities and the results inside each object are smoother.

In Fig. 5 we show the angular error for every frame of the sequence – for convenience we only show the central frames –. We may appreciate that the spatial method provides worse and less stable results. Nearly all the results for the temporal methods are improved. In all cases, the "Bi-Temporal" performs in a similar way as the "Temporal" but with a smaller magnitude.

As we could expect, the results obtained for this sequence are improved. This sequence is similar to the translating square in the sense that the objects are



Fig. 3. Frames 0, 10 and 20 of the Marble Blocks sequence*

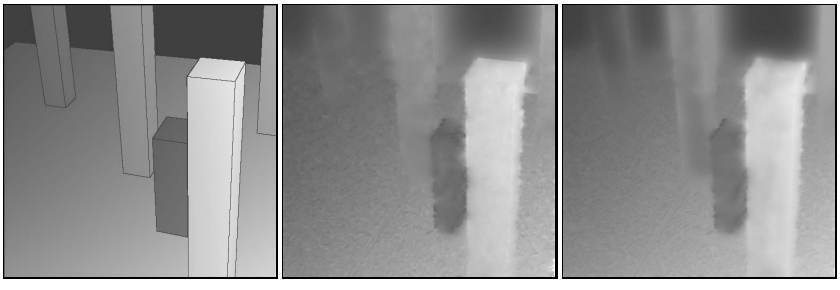


Fig. 4. On the left, the true disparity associated with frame 10; in the middle, the optical flow corresponding to the spatial method; and, on the right, the solution for the bidirectional temporal method

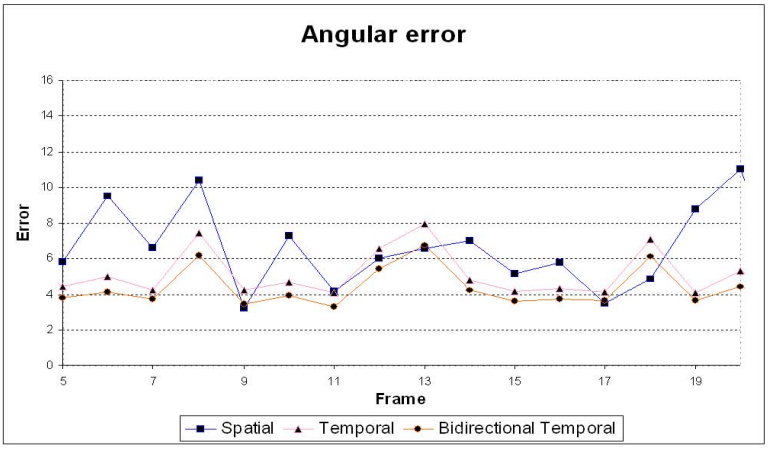


Fig. 5. Angular error for the Marble Block sequence

* This sequence is composed of 30 frames and is copyright by H.-H. Nagel KOGS/IAKS, University of Karlsruhe, Germany, at http://i21www.ira.uka.de/image_sequences/.

Table 2. Mean angular and euclidean errors for the Marble Block sequence

Method	AE_μ	AE_σ	EE_μ	EE_σ
Spatial	$6{,}695^o$	$2{,}698^o$	0,2480	0,0963
Temporal	$5{,}402^o$	$1{,}327^o$	0,2081	0,0638
Bi-Temporal	$4{,}731^o$	$1{,}330^o$	0,1848	0,0661

moving with constant velocity in the same direction. In fact, the average motion of the sequence is 1,33 – with standard deviation 0,03 – except for frames 4, 8, 13, 18, 23 and 28 in which the average motion is 1,58 – with standard deviation 0,02–. This means that there is a shift on the magnitude of the velocities at these frames. If we look at Fig. 5 we observe maximum peaks that reduce the stability of the temporal methods. We suppose that this sequence was captured with different frame rates at these particular frames.

In table 2 we show the angular and euclidean errors and their standard deviations. The improvement of the "Temporal" method with respect to the "Spatial" one is about 19,31% for the angular error and 12,05% for the euclidean error. The improvement of the "Bi-Temporal" is about 29,33% for the angular and 25,48% for the euclidean error. In both cases the standard deviations are considerably reduced in a similar magnitude.

5 Conclusions

With this work we have proposed a novel variational method that includes a temporal smoothness term. We have created a new term that minimizes the difference of the optical flows that permits large displacements of the objects.

We have shown the necessity of including the backward flow in the energy functional. All the comparisons show that the bidirectional temporal method clearly outperforms the simpler temporal method.

As we can observe in the experimental results, the numerical studies show that this method provides more stable results. We have also shown that when translating motions are present on the scene then there is an increase of accuracy in all the frames.

To our knowledge, this is the first approach that considers temporal regularizations with large displacements.

References

1. Alvarez, L., Deriche, R., Papadopoulo, T., Sánchez, J.: Symmetrical dense optical flow estimation with oclussions detection. International Journal of Computer Vision. Preprint (2006)
2. L. Alvarez, J. Weickert, and J. Sánchez, Reliable Estimation of Dense Optical Flow Fields with Large Displacements. International Journal of Computer Vision, 391 (2000) 41–56. An extended version maybe be found at Technical Report n^{o}2 del Instituto Universitario de Ciencias y Tecnologías Cibernéticas

3. Anandan, P.: A Computational Framework and an Algorithm for the Measurement of Visual Motion. International Journal of Computer Vision 2, 283–310 (1989)
4. Black, M.J., Anandan, P.: Robust dynamic motion estimation over time. In: Proc. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 292–302 (1991)
5. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High Accuracy Optical Flow Estimation Based on a Theory for Warping. In: Pajdla, T., Matas, J. (eds.) ECCV 2004. Part IV. LNCS, vol. 3024, pp. 25–36. Springer, Heidelberg (2004)
6. Cohen, I.: Nonlinear Variational Method for Optical Flow Computation. In: Proc. of the 8th Scandinavian Conference on Image Analysis, Norway (1993)
7. Deriche, R., Kornprobst, P., Aubert, G.: Optical flow estimation while preserving its discontinuities: a variational approach. In: Proc. Second Asian Conference on Computer Vision, vol. 2, pp. 290–295 (December 1995)
8. Horn, B., Schunck, B.: Determining Optical Flow. Artificial Intelligence 17, 185–203 (1981)
9. Mémin, E., Pérez, P.: Dense Estimation and Object–Based Segmentation of the Optical Flow with Robust Techniques. IEEE Transactions on Image Processing 7(5), 703–719 (1998)
10. Nagel, H.H.: Extending the 'oriented smoothness constraint' into the temporal domain and the estimation of derivatives of optical flow. In: Faugeras, O. (ed.) Computer Vision - ECCV 90. LNCS, vol. 427, pp. 139–148. Springer, Heidelberg (1990)
11. Nagel, H.H., Enkelmann, W.: An Investigation of Smoothness Constraints for the Estimation of Displacements Vector Fields from Image Sequences. IEEE Trans. Pattern Anal. Mach. Intell. 8, 565–593 (1986)
12. Wells, W.M., Viola, P., Atsumi, H., Nakajima, S., Kinikis, R.: Multimodal Volume Registration by Maximization of Mutual Information. Medical Image Analysis 1, 35–51 (1996)
13. Weickert, J., Schnörr, C.: Variational Optic Flow Computation with a Spatio–Temporal Smoothness Constraint. Journal of Mathematical Imaging and Vision 14, 245–255 (2001)

Comparing Improved Versions of 'K-Means' and 'Subtractive' Clustering in a Tracking Application

Marta Marrón Romera, Miguel Angel Sotelo Vázquez, and Juan Carlos García García

Electronics Department, University of Alcala,
Edificio Politécnico, Campus Universitario Alcalá de Henares, 28871, Madrid, Spain
{marta,sotelo,jcarlos}@depeca.uah.es

Abstract. A partitional and a fuzzy clustering algorithm are compared in this paper in terms of accuracy, robustness and efficiency. 3D position data extracted from a stereo-vision system have to be clustered to use them in a tracking application in which a particle filter is the kernel of the estimation task. 'K-Means' and 'Subtractive' algorithms have been modified and enriched with a validation process in order improve its functionality in the tracking system. Comparisons and conclusions of the clustering results both in a stand-alone process and in the proposed tracking task are shown in the paper.

Keywords: clustering, probabilistic-deterministic, particle-filters, tracking.

1 Introduction

Clustering algorithms are used in a large amount of applications ([1], [2], [3]). In artificial vision, these processes are particularly useful as compress visual data generating classes that contain more accurate and robust environmental information.

In the application developed by the authors in [4], information about the 3D position of a variable number of objects in an indoor environment is obtained from a stereo-vision system and used as measurement vector of a probabilistic estimation process, in a multi-obstacle detection and tracking process. This information is clustered in 2D before using it in the probabilistic tracker, as it is explained in [4].

It has been proven ([5]) that an association algorithm is needed in order to make robust the multi-tracking process. Most of the association solutions are based on the Probabilistic Data Association (PDA) theory [6], such as the Joint Probabilistic Particle Filter (JPDAF) like in [7], but the computational load of these techniques is generally a problem for the real time execution of the tracking algorithms. On the other hand, some of the solutions presented by other authors ([8]) lack of robustness if the measurements used in the estimation process are not wisely preprocessed. The use of a clustering process solves the association problems.

Fig. 1 includes a flowchart describing the global tracking application. In the same figure, an image showing the 3D position data extracted from the objects to track in a real experiment is also included, as white dots.

It has to be remarked that, though 3D position information of the objects to be tracked is available, the height coordinate of these data is not useful in the clustering, and thus, the position measurements are cluster in a 2D space.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 717–724, 2007.
© Springer-Verlag Berlin Heidelberg 2007

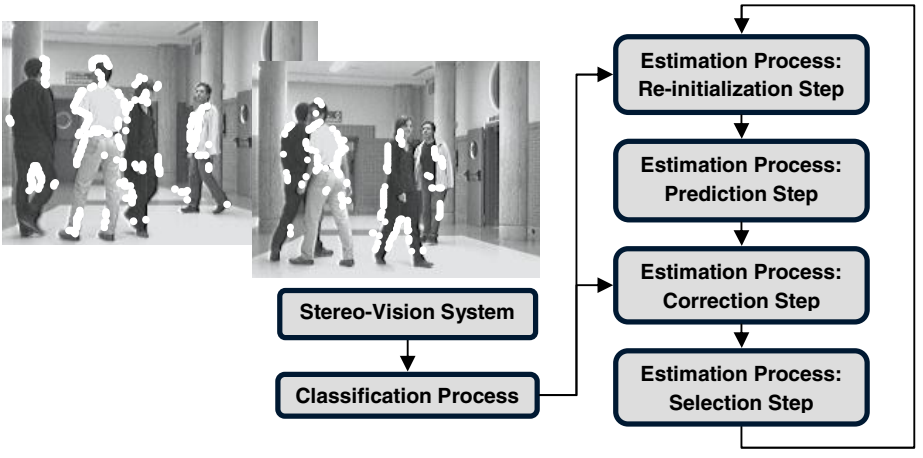


Fig. 1. Functional description of the multimodal estimation tracker, and the clustering process implication in it. A frame extracted from a real experiment is also included. In the image the position data points to be clustered are plotted in white.

On the other hand, as it can be notice in Fig. 1, position measurements extracted by the stereo-vision system are not equally distributed among all objects in the environment, and hence, the tracking algorithm, and what is of interest in this paper, clustering proposals should be able to manage clusters with very different likelihood.

In this paper, two different clustering algorithms are tested and compared in order to find the most adequate solution for the tracking task exposed. The contributions of the clustering proposal to the multi-tracking process are also shown in the paper.

2 Clustering Algorithms

As mentioned two different clustering algorithms are compared in their application to the objective pursuit. They are described in the following paragraphs:

- *Modified 'K-Means':* The deterministic solution presented by MacQueen in [9], and afterwards improved ([10]), is modified to classify a variable number of clusters. A flowchart of this clustering proposal is presented in Fig. 2, in which the functionality included to achieve an adaptive behavior of the algorithm to the number of clusters generated by the algorithm is marked with a dashed line. On the other hand, in order to improve the algorithm speed in its recursive performance ([11]), the 'K-Means' segmentation process starts looking for clusters centroides near the predicted values that are calculated with the clustering results in the previous execution of the algorithm. This other improvement is also shown in dashed line in Fig. 2.
- *Modified 'Subtractive':* This algorithm is a fuzzy solution for the clustering task based on the Mountain algorithm [12], and presented afterwards in [13]. The 'Subtractive' algorithm is unlinked, and therefore a process to obtain the each cluster centroide and the members has to be included in the standard algorithm if

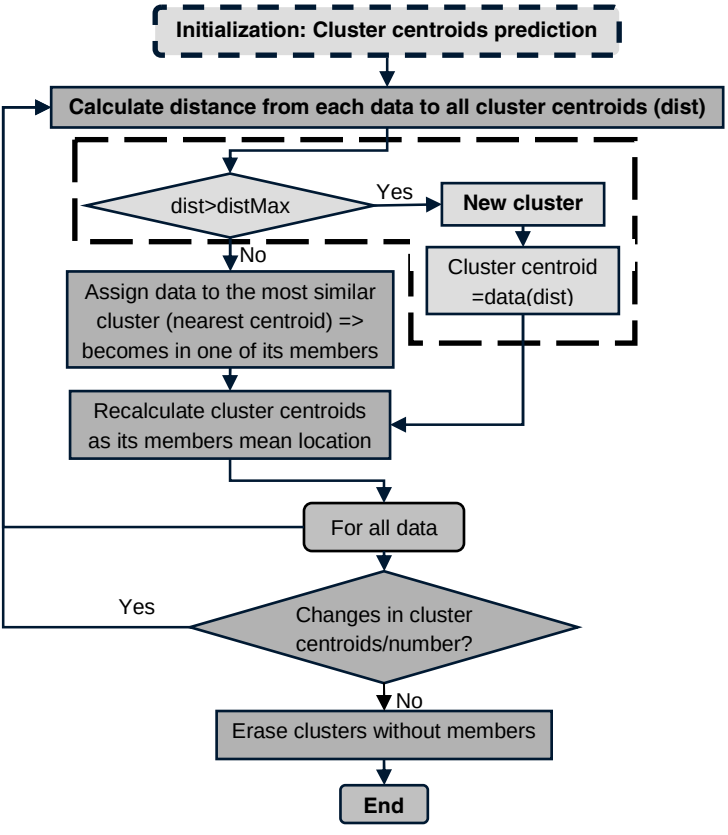


Fig. 2. Flowchart of the modified 'K-Means' proposed by the authors. Dashed lines remark the main improvements included in the standard version of the algorithm.

needed, as it is in the present application. This improved functionality is shown with dashed lines in Fig. 3. Moreover, the probability of each cluster members is reset in order to decrease the cluster duplication error rate of the standard fuzzy classifier. This process is also included in Fig. 3 with dashed lines.

All the modifications described increase the efficiency and accuracy of the basic classifiers. To improve their robustness, a validation process is also added to both clustering algorithms. This procedure is useful when noisy measurements or outliers produce a cluster creation or deletion erroneously. In these situations none of the modified algorithms shown in previous figures behave robustly. The validation process is based in two parameters:

- *Distance between the estimated and the resulting cluster centroide.* The clusters centroide estimation process in the modified 'K-Means' is also developed for the 'Subtractive' version of Fig. 3. Estimated centroides are then compared within two consecutive iterations, to obtain a confidence value for clusters validation.

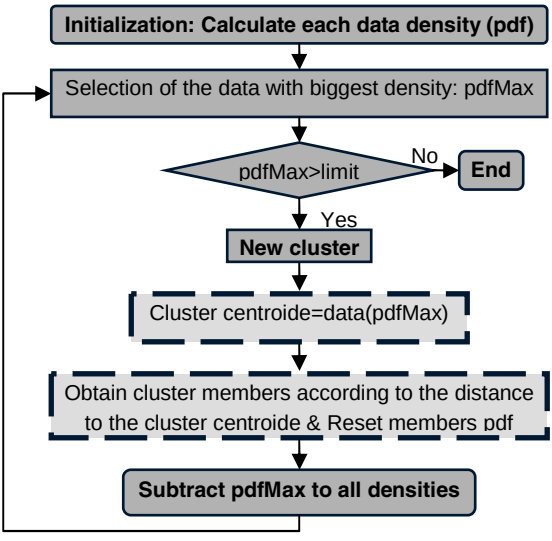


Fig. 3. Flowchart of the modified 'Subtractive' algorithm proposed by the authors. Dashed lines remark the main improvements included in the standard version of the algorithm.

- *Cluster likelihood.* The clusters probability information inherent in 'Subtractive' algorithm is also calculated in 'K-Means', as a function of the data agglomeration in each cluster. This likelihood value is also used as a validation or confidence parameter in the cluster validation process.

In the following sections the results obtained with the clustering proposals described are deeply commented.

3 Comparison

In order to extract comparative conclusions about the accuracy, robustness and efficiency of the clustering methods presented, they have been run with different data sets containing 2D position measurements obtained from the objects in the environment during the tracking task.

The objective in all situations is that a cluster is created for each object in the scene by the algorithms if some data points related to it are present in the clustered data set. The main conclusions extracted from the global set of experiments (2931 frames of 3 situations of different complexity) are the following:

- In general 'K-Means' shows higher reliability than 'Subtractive', with a mean error rate of 3.9% against 6.7%.
- High accuracy is obtained in the whole testing process, since less than a 1% of errors is due to shifts if 'K-Means' is used, and 1.6% if it is 'Subtractive'.
- The most recurrent error (half of the global error rate with both algorithms) appears when an object is poorly measured and only a few data points related to it are included in the set. This type of error is generally called a missing error.

- The generation of a duplicated class related to the same object is a typical error generated by 'Subtractive' (1.2% error rate against less than 1% in 'K-Means'). Nevertheless this error rate is doubled if the reset process included as an improvement in the 'Subtractive' algorithm is not used.
- The validation process proposed rejects almost the 90% of noisy measurements when included in any of the clustering algorithms. In any case, 'Subtractive' clustering behaves better with noisy measurements than 'K-Means' as almost 50% of the global error rate when using 'K-Means' is due to noise.
- 'K-Means' is less time consuming than 'Subtractive' (a mean execution time of 1.5ms for 'K-Means' versus 17ms for 'Subtractive'[1]). The execution time of 'K-Means' is decreased almost in a 50% if using the centroide estimation process.

Table 1 shows the main errors generated by both clustering proposals in a complex experiment of 1054 frames, in which situations of until 6 static and dynamic classes are included. The results displayed in the table validate the comments exposed in previous paragraphs and confirm the better behavior of the clustering proposal based on the standard 'K-Means'.

Table 1. Rate of different types of errors obtained with the proposed versions of 'K-Means' and 'Subtractive' algorithms in a 1054 frames experiment of none until 6 classes

	K-Means (% frames with error)	Subtract (% frames with error)
Missing	10.9	19.3
Duplicated	6.1	16.2
Shift	0	9.7
2 as 1	3.9	2.3
Total	**20.9**	**47.4**

Fig. 4 shows the results of two sequential iterations of the clustering proposals in the complex experiment mentioned. Numbers at the top and at the bottom of the images indicate the number of clusters generated by 'K-Means' and 'Subtractive', respectively. Regular rectangles display the clusters generated by 'K-Means', and dashed the ones generated with 'Subtractive'. In the image of the right, the shift error generated by 'Subtractive' clustering can be observed in person tacked with number 3. In the left image, a missing error generated by 'Subtractive' with person number 4 that is partially occluded by person number 2, is shown. Results generated by 'K-Means' clustering are right in both cases.

The error rate is, in any case, too high to use the output of the clustering process as output of the multi-tracking process. The difference in the likelihood of the measurements sensed by the stereo-vision system for each object in the environment cannot be properly handled by the clustering proposals. The use of a probabilistic estimator that reinforces the probability of poorly sensed objects is therefore justified.

[1] The algorithms are run in an Intel Dual Core processor at 1.8GHz with 1Gbyte RAM.

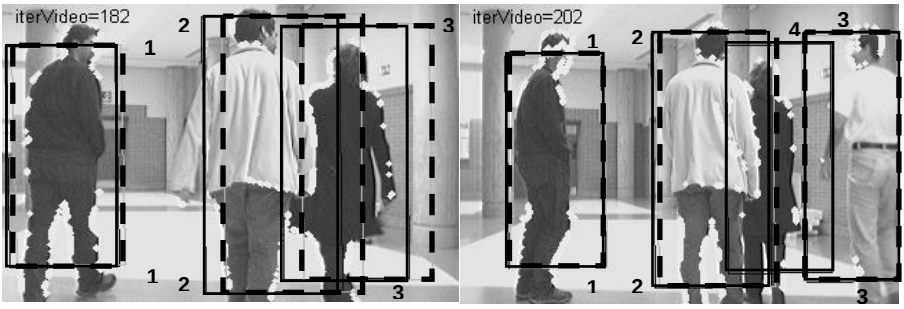


Fig. 4. Two sequential images extracted from a complex experiment. Clustering results are shown with regular rectangles for the 'K-Means' proposal and dashed for the 'Subtractive' one.

4 Application of the Clustering Algorithms to the Tracking Process

As mentioned in section 1, the clustering algorithms developed were designed to be used in a multi-tracking application ([4]), in order to increase the tracker robustness.

A particle filter (PF) is used as position estimator, and the algorithm is extended and the clustering process is used as association process in order to achieve the desired functionality and efficiency of the multi-tracker. The final algorithm is called 'Extended Particle Filter with Clustering Process' (XPFCP).

The functionality of the clustering algorithms in the tracking process is analyzed in detail in [4]. The general idea is to use the process in two different steps of the multimodal estimator (see Fig. 1) as follows:

- *At the re-initialization step*, position information compressed within the clusters is used to create new hypotheses in the multimodal estimator. The clustering process is thus used to adapt the tracker functionality to situations with a variable number of objects.
- *At the correction step*, the cluster probability is used to reinforce or weaken each object position estimation hypothesis. This procedure increases the robustness of the global tracking task.

Both algorithms have been used within the probabilistic tracker mentioned, to test which one is the best solution to increase the multimodal estimator robustness.

Table 2 summarizes the results obtaining when comparing both clustering proposals in the XPFCP designed, in the same 1054 frames experiment used to extract results in Table 1. As it can be notice in the table, the solution based on 'K-Means' has better performance than the one based on 'Subtractive'.

In a more general set of tests (2931 frames of 3 situations of different complexity) the results are similar to those shown in Table 2:

- The mean error rate is 1.8% if using 'K-Means' and 3.7% if using 'Subtractive' in the multi-tracking algorithm. Moreover, only missing errors appear if using the 'K-Means' algorithm, while the 'Subtractive' proposal shows also duplicating errors.
- The mean execution time of the multimodal tracker is 19.1ms when using 'K-Means', almost four times lower than if 'Subtractive' is used (73.3ms). The reason

for this difference is that the execution time of the estimation process without clustering is comparable to the one of the stand alone version of 'Subtractive' (a mean value of 17.8ms).

- Outliers do not appear in the tracker results, so the problem of noise has been erased from the tracking application, thanks to the clustering process.

Table 2. Rate of different types of errors obtained with the XPFCP proposed in a 1054 frames tracking experiment of none until 6 classes

	K-Means (% frames with error)	Subtract (% frames with error)
Missing	12.9	16.4
Duplicated	0	4.7
Total	**12.9**	**21.1**

All exposed results recommend the use of 'K-Means' in the multi-tracking process. An example of the results achieved with the XPFCP is shown in Fig. 5, where the modified 'K-Means' is used as association process in the multi-object tracker. Tracked objects are displayed with rectangles in the images. This figure proofs the correct functionality of the global proposed tracker also in crowd situations where multiple objects cross and are partially occluded during various iterations.

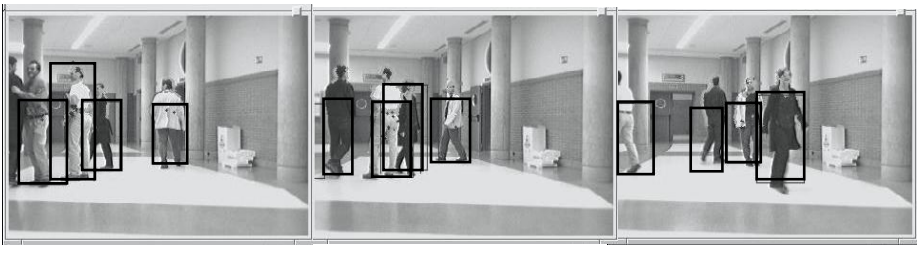


Fig. 5. Three frames in a real tracking sequence showing results generated by the XPFCP in the tracking process. Clusters are plotted with rectangles.

5 Conclusions

In this paper two new clustering proposals based on the standards 'K-Means' and 'Subtractive', are presented. The improved algorithms are thought to be applied in the multi-object tracking system described by the authors in [4], called XPFCP. The clustering process is used as association algorithm in the multimodal estimator. On the other hand, the use of clustered measurements increases the tracker robustness.

The two clustering proposals performance is compared and some interesting conclusions are extracted:

- The proposal based on 'K-Means' shows higher reliability than the one based on 'Subtractive'.

- 'Subtractive' behaves better with noisy measurements than 'K-Means', but the validation process included in both clustering proposals solves successfully the robustness problems generated by outliers.
- The execution time of 'Subtractive' is almost 10 times bigger than the 'K-Means' one, and comparable to the multimodal estimator one.

All these conclusions recommend the use of the modified 'K-Means' in the XPFCP, and the success of the resultant multi-tracking algorithm is also demonstrated in the paper.

Acknowledgments. This work has been financed by the Spanish administration (CICYT: DPI2005-07980-C03-02).

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(nº 3), 264–323 (1999)
2. Berkhin, P.: Survey of clustering data mining techniques. Technical Report (2002)
3. Everitt, B., Landau, S., Leese, M.: Cluster analysis. Edward Arnold Publishers, 4th Edition, London, ISBN: 0-340-76119-9 (2001)
4. Marrón, M., Sotelo, M.A., García, J.C., Fernández, D., Pizarro, D.: XPFCP: An extended particle filter for tracking multiple and dynamic objects in complex environments. In: Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS05), Edmonton, pp. 234–239 (2005), ISBN: 0-7803-9252-3
5. Isard, M., Blake, C.A.: Conditional density propagation for visual tracking. International Journal of Computer Vision 29(nº1), 5–28 (1998)
6. Bar-Shalom, Y., Fortmann, T.: Tracking and data association. Mathematics in Science and Engineering, vol. 182. Academic Press, London (1988)
7. Schulz, D., Burgard, W., Fox, D., Cremers, A.B.: People tracking with mobile robots using sample-based joint probabilistic data association filters. International Journal of Robotics Research 22 (nº 2), 99–116 (2003)
8. Koller-Meier, E.B., Ade, F.: Tracking multiple objects using a condensation algorithm. Journal of Robotics and Autonomous Systems 34, 93–105 (2001)
9. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, vol. 1, pp. 281–297 (1967)
10. Pelleg, D., Moore, A.: X-means: Extending k-means with efficient estimation of the number of clusters. In: Proceedings of the Seventeenth International Conference on Machine Learning (ICML00), San Francisco, pp. 727–734 (2000) (ISBN: 1-55860-707-2)
11. Weiss, Y.: Belief propagation and revision in networks with loops. Technical Report 1616, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (1997)
12. Yager, R.R., Filev, D.: Generation of fuzzy rules by mountain clustering. Journal of Intelligent and Fuzzy Systems 2(nº 3), 209–219 (1994)
13. Chiu, S.L.: Fuzzy model identification based on cluster estimation. Journal of Intelligent & Fuzzy Systems 2(nº 3), 267–278 (1994)

FPGA Real Time Lane Departure Warning Hardware System

Pedro Cobos Arribas and Felipe Jiménez Alonso

Instituto Universitario de Investigación del Automóvil (INSIA) Universidad Politécnica de Madrid (UPM)Campus Sur UPM. Carretera de Valencia km 7 28031 Madrid
pedro.cobos@upm.es, felipe.jimenez@upm.es

Abstract. This paper presents a design adapting the Kalman Filter to the vehicle system domain and Field Programmable Logic technology. The objective to which the system will be applied is detection of road lines from visual information, derived from a low cost monochrome camera with real time response requirements and good results for real scenarios (secondary roads, rain, damaged or occluded road lines..). The sections will describe how the original algorithm is mapped to a real time hardware vision system, which includes a low-cost FPGA processing system and a camera, for vehicle applications. The paper will also illustrate how the needed tasks have been implemented on the FPGA, with the logical architectural restrictions. It mentions also the ways in which overall performance will be increased.

Keywords: Vehicle safety, Real time, Low cost, hardware system, FPGA, Kalman filtering.

1 Introduction

The objectives of the investigation are detailed. It is cited the great interest of vision applications in the automotive industry specially to avoid car accidents. Vision capabilities in cars will improve the safety of drivers and pedestrians. This sounds quite good, but actually are few applications on standard cars. In this line a robust and inexpensive system is proposed.

First of all, the global system structure adopted in all active vision applications is showed at Fig.1.

All the actions, in a robotic environment or on a automobile one, are motivated from the vision sensor(s), and with the aim to develop the process in real time, and with low cost restrictions.

In the following sections, some examples of vision algorithms previously developed are described, as well as their advantages, troubles and potential applications. The next part contains the experimental description examples of the hardware vision system. Finally, in last section, the results and conclusions are presented

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 725–732, 2007.
© Springer-Verlag Berlin Heidelberg 2007

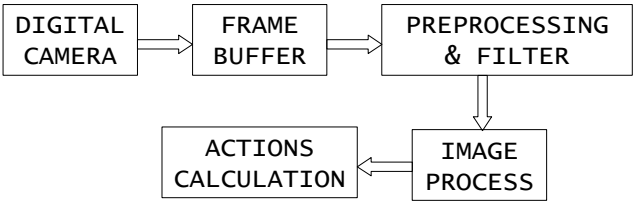


Fig. 1. Diagram of the active vision architecture

2 Previous Work

In this section some implementations of vision systems, previously developed, are cited and described to show its approaches, troubles and adopted solutions. In [1-4] several systems are described. The structure of all of them is the same (Fig.2), where an easy interconnection an real time operation are provided. It is obvious that a little hardware block can work quickly than a large one, and many of these blocks can operate in a parallel mode if the intermediate results are accessible. In this way, the input and output module memories can be accessed at the same time from the source and the target systems and are the more relevant elements of this hardware structure. The other processing element is always a FPGA, that holds the vision algorithm (normally very intensive mathematical calculation) and the memories addressing tasks.

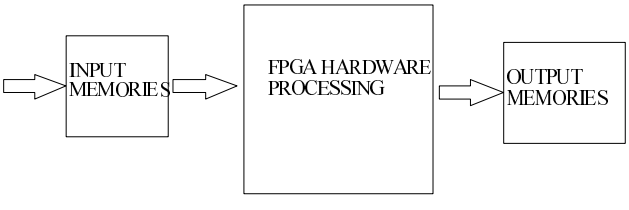


Fig. 2. System structure

Two examples of tipical motion moudules are shown at Figs 3 and 4. Fist one is an correlation optical flow algorithm, that looks for the image movement in blocks of 16 image lines of 96 pixels from 480 possible ones, stored in dual port memories. The FPGA used here was an ALTERA EPF10K50 and the process rate obtained was o 22,56 images (96x98 pixels sized) per second.

The second one is a compression unit that employs Log Polar algorithm to reduce the data image size from 256Kb of the original images to 9,6Kb of the log-polar images. In this example, the size image and speed of operation with an ALTERA epf8282, were 25 images (96 radios of 128 points) from the original 480x500 pixels. Others implementation of optical flow algorithms were made, all of them with real time results.

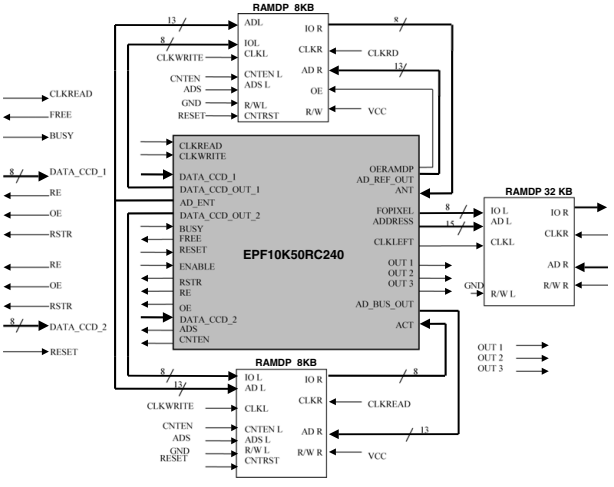


Fig. 3. Blocks Diagram of the Camus Correlation Optical flow Algorithm

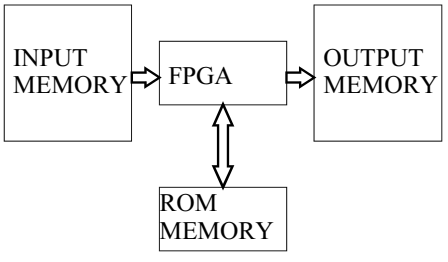


Fig. 4. Blocks Diagram of the Log Polar FPGA Implementation

Only were one serious problem: every implementation needed a different printed circuit board with a different PFGA and memories. To solve this problem, a reconfigurable system, described next, was developed.

3 Hardware Vision System Description

The components and structure of the vision system used in this application are described. It is a modular and versatile system that processes the information from a digital camera to some frame memories and from them to a FPGA. The output of one system can be connected to another one. In this way it is possible to make some parts of an algorithm, that works in a parallel way. The blocks diagram and board views are shown at Figs 5-6, and the operation of the system is fully described at [9],[10]. Previous hardware implementations can easily be repeated with this system, due to its flexibility. So new ones will that will be

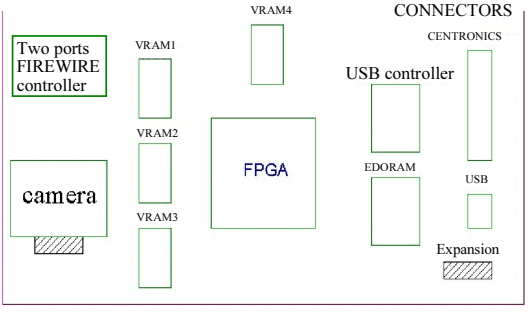


Fig. 5. Blocks Diagram of the hardware vision system



Fig. 6. Top and bottom sides of the board

developed soon. The board dimensions and power consumption are little, so it can be assembled on real systems, as robots or vehicular controllers.

The vision system has the following input options:

- A low-cost, monochrome, digital camera (OV7120) with a 640 x 480-pixel resolution, which can be directly connected to the vision board using a 32-pin flat cable female connector.
- An IEEE-1394 camera, which can be connected using a Firewire input connector. This is for applications with more features and higher costs. This option allows the interweaving of the vision system in applications that require the processing of images that reach the camera at high speeds.
- Another identical block output to break down complex processing into several phases. As a result, each system runs in parallel parts of the operations, and a complex algorithm can be completed using simple blocks, with each block hosted on a vision board.

The above input options can be connected directly to the FPGA. However, they are normally connected to several memories (in this application three input memories) in order to store consecutive image information. These memories are frame buffers, that is, memories used to store a full image, of sequential access, especially the AL422, with two stand-alone input/output ports. This is so that they can be written and read at different speeds. They have a capacity of 3Mbits and therefore the images can be easily stored in different formats. The processing outputs can be obtained from several connectors:

- A flat cable male connector, which is the male version of the female input connector, to be able to connect several systems in parallel so that the system outputs to the connector or the output memory. It is configured with a signal organization identical to the one in the system input.
- An USB connector to monitor the processing output from a PC. To do this, it is necessary to have a specific controller (OV511+) from the same manufacturer as that of the cameras.
- A Firewire connector to read the results from this format. Alongside these input/output connections, another 10-pin connector is required for the programming of the FPGA from a PC, through a cable connected to its serial port. This programming can be stored in a non-volatile configuration memory, included in the system, so that there is no need to re-programme every time the power is shut down. In Figure 7, you can see the layout of a camera connected to the vision system, and of two systems operating from a camera. These are examples of the system operational modes.

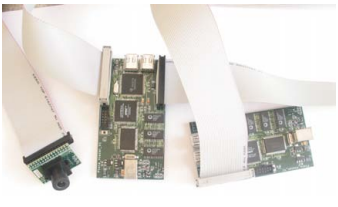
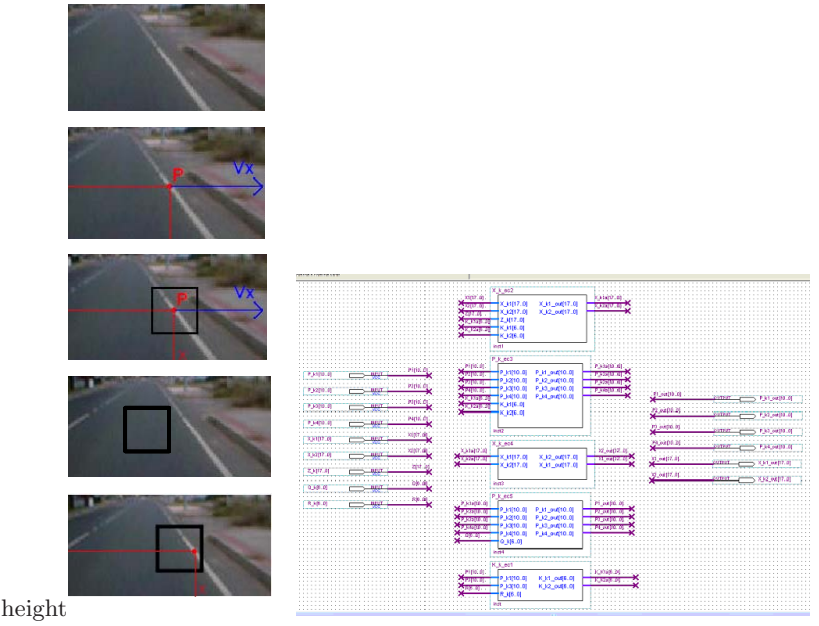


Fig. 7. Details of the interconnection between two vision systems and a camera, using flat cable connectors

4 Kalman Implementation Details

The blocks required, Sobel operator, preprocessing blocks, the simulations and the hardware solutions used are explained in this point. All of the components have been exhaustively simulated with Matlab, and a VHDL description of every one of them is provided. Next figure (Fig.8) shows at the left the phases of the algorithm, and at the right a QUARTUS block diagram with the for main hardware components of the circuit. Some useful references can be found in [8].

These blocks are completelly simulated and some of the output results can be shown an Fig.9.



height

Fig. 8. Implementation of Kalman filter



Fig. 9. Traffic sequences

They are frames of a sequence that shows how the Kalman predicted lane centroid is correctly tracked over four road scenarios, with real illumination conditions. The images have been obtained from a low cost camera on a car in several proofs.

5 Conclusion and Future Developments

The results that are being obtained now shows that the Kalma Filter unit spends less than 10 ms in its execution. This is a real time and cheap solution. The original contribution of this work is the building of a modular hardware architecture, which allows the interconnection of the different modules. The hardware implementations of vision algorithms will be very important in real applications. This is due to the possibility of having large complex ASIC, and larger and faster FPGA, together with the gradual sophistication and increase of the capacity of the all-purpose processors or the DSP processors. The research done intends to cover an area of the important developments in hardware for artificial vision. Therefore, this research does not intend to be a new application for artificial vision, but a hardware architecture that allows the implementation of vision algorithms by trying to generate vision systems that include and leverage different hardware elements, as real-time special solutions. These will complement the typically programmed systems. During this research, we have found that many of the techniques based on the calculation of the optical flow, like the ones used here, are very limited in real-time applications, where it is necessary to reduce the dependency on the restrictions derived from the projection process at the image level. However, by reaching the processing speeds indicated and using elements accessible to any researcher (such as the FPGA) the work will be relevant. This is especially true in robotic applications where the vision is used by the robots to avoid obstacles, follow paths or to perform any other required task. Another application where vision systems could be key is in the development of help systems for driving vehicles to avoid collisions with other vehicles or other types of road traffic accidents. The future development of this investigation can be summarized in the inclusion of the developed vision module in a robotic head with motors to control the camera(s) with three degrees of free movement, with biological like action. Use of the vision module on motorized vehicles so that their movement can be controlled according to the visual information received. Development of other solutions that include digital signal processors (DSP) and programmable logic devices in order to create other vision applications. This new hardware system would perform functions that require intensive-data processing with one or several FPGAs, and mathematical functions, or data-processing, with DSP processors. Many of the future results depend greatly on the electronic technology evolution, which in coming years will be able to provide integration levels, which were inconceivable in recent times, including microprocessors with a high-level parallelism, and with high-capacity and high-speed FPGAs. We believe that with this fast calculation potential, vision algorithms and dedicated

hardware will be developed and improved so that tasks that are apparently simple (such as obtaining optical flow to be able to generate the structure of an image from movement data) stop being an extremely complex issue [7], which vision scientists have been investigating in the last 25 years.

References

1. Cobos, P., Monasterio, F.: Fpga implementation of the Horn & Schunk Optical Flow Algorithm for Motion Detection in real time Images. Dcis'98. In: Proceedings, XIII Design of circuits and integrated systems conference, pp. 616–621 (1998)
2. Cobos, P., Monasterio, F.: Fpga implementation of a Log-polar Algorithm for real time Applications. Dcis'99. Proceedings, XIV Design of circuits and integrated systems conference, 63–68 (1999)
3. Cobos, P., Monasterio, F.: FPGA implementation of Camus correlation Optical Flow Algorithm for real time images. In: Beauchemin, S.S., Nouboud, F., Roth, G. (eds.) Vision Interface Proceedings VI2001, 14th International Conference on Vision Interface, Canada, pp. 7–9 (2001)
4. Cobos, P., Monasterio, F.: FPGA: implementation of Santos_Victor optical flow algorithm for real_time image processing: an useful attempt. In: Proceedings of SPIE, vol. 5117, pp. 23–32 (2003)
5. Cobos, P., Monasterio, F.: FPGA Based Development Vision System. Dcis 2001. In: Proceedings, XVI Design of circuits and integrated systems conference, pp. 322–326 (2001)
6. Cobos, P., Monasterio, F.: FPGA Board For Real Time Vision Development Systems. In: Proceedings of the ICCDCS 2002, 4th IEEE International Caracas Conference on Devices, Circuits and Systems, IEEE Computer Society Press, Los Alamitos (2002)
7. Chellapa, R., Quian, G., Srinivasan, S.: Structure from motion: sparse versus dense correspondence methods. In: IEEE Proceedings 1999 International Conference on Image Processing, vol. 2, pp. 492–499 (1999)
8. Paul, A.S., Wan, E.A.: Dual Kalman filters for autonomous terrain aided navigation in unknown environments. IJCNN '05. In: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, pp. 2784–2789. IEEE Computer Society Press, Los Alamitos (2005)

Efficient Combination of the Fuzzy Hough Transform and the Burns Segment Detector

Marta Penas[1], María J. Carreira[2], Manuel G. Penedo[1], and Noelia Barreira[1]

[1] Dpto. Computación. Fac. Informática, Universidade da Coruña. 15071 A Coruña. Spain
{mpenas,mgpenedo,nbarreira}@udc.es
[2] Dpto. Electrónica e Computación. Universidade de Santiago de Compostela. 15782 Santiago de Compostela. Spain
mjose@dec.usc.es

Abstract. This paper describes a computational framework for the fully automated detection of line segments in 2D digital images. The operation of the framework is divided in two stages, the low level directional primitive detection through Gabor wavelets and growing cell structures, and the segment detection through an efficient and very accurate combination of the fuzzy Hough transform and the Burns segment detector.

Keyword: Gabor wavelets, growing cell structures, chromaticity diagram, Hough transform, Burns segment detector.

1 Introduction

The boundaries of the objects in an image often lead to oriented and localised changes in the intensity called edges. The edge and segment detection are the first stages in many image analysis applications and they are of great importance as they constitute the basis for the higher stages, with a great influence on the final performance.

It is very difficult to design a general low-level detector capable of dealing with any kind of input images and fulfil the requirements of the subsequent processing stages. For this reason, many low-level detection algorithms [8,12,9] have been implemented over the last years, being the Canny edge detector [2] the most popular of them. Similarly, the Hough transform [7], with its multiple variants, is also the most popular line segment detector in the literature [14,1].

The operation of most of the edge and segment detectors in the literature is governed by one or several parameters that must be adjusted by the final user according to the input image features. The aim of the work described in this paper will be to describe a computational framework suitable for the analysis of a wide variety of input images without any final-user parameter tuning. The framework is divided into two stages: (a) low-level directional primitive extraction and organisation, and (b) segment detection. We will focus our attention on the final stage of the framework, the segment detection, that has been implemented through an efficient combination of the fuzzy Hough transform [7] and

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 733–739, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the Burns segment detector [1]. The methodology for the combination of the fuzzy Hough transform and the Burns segment detection is the main contribution of the present paper.

This paper is organised as follows. Sec. 2 describes the low level directional primitive extraction, implemented through the Gabor wavelet decomposition [6] and a growing cell structure [4], sec. 3 describes the segment detection stage and 4 contains the conclusions from our work.

2 Directional Primitive Extraction

As previously mentioned, in the computational framework described, the low level directional primitives are extracted through the Gabor wavelet decomposition [6] and organised through a growing cell structure [4].

The Gabor wavelets [6] are complex exponential signals modulated by Gaussians with two important properties that make them good edge detectors: the optimal localisation of the edges in the space [3] and the absence of image-dependent parameter tuning. Their most important drawback is their greedy demand in memory and computational time. For this reason, in a previous paper [10], we have implemented an efficient, multi-resolution spatial domain implementation of the Gabor wavelet decomposition based on the convolution of the input image with 1D masks obtained from the decomposition of the 2D masks that define the wavelets. This implementation makes use of the good edge localisation property of the Gabor wavelets, with the exact position of an edge determined as a conjunction between the maximum in the modulus and a zero crossing in the even or the odd part of the Gabor results.

The computational framework in [10] makes use of a bank of 8 Gabor wavelet filters centred at frequency $\frac{1}{4}$ and eight equidistant orientations $\frac{k\pi}{8}, k = 0, ..., 7$ to filter the input image, leading to 8 result images. The integration of these images into a unique result is necessary for the sake of the efficiency and is accomplished through a growing cell structure (GCS) [4], this is, a dynamic neural network that allows the insertion of new processing elements during the operation of the structure in order to adequately cover the input space.The input to the growing cell structure is a nine component vector assigned to each input pixel (x, y), composed of the eight Gabor decomposition results and a ninth component, named *response*, that takes a value in $[0, 1]$ and determines the presence of a directional feature. The first eight components are scaled such that their modulus is equal to the *response* in order to reduce the inter-image variability.

In order to graphically represent the results of the growing cell structure, each processing element was assigned a colour from a colourmap to indicate its orientation. The colourmap was built from the maximum circle inside the RGB triangle in the chromaticity diagram[13], as depicted in fig. 1 left. Fig. 1 right shows the colours assigned to the entire direction space.

The low-level directional primitive detector just described has been applied to wide variety of input images, three of them depicted in fig. 2. As these images

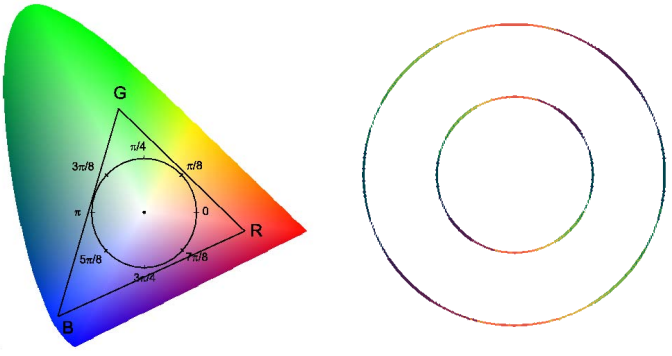


Fig. 1. Left: Colourmap inside the RGB triangle. Right: Colour gradation in the direction space.

show, the detector is capable of detecting the most salient features in the input images while removing most of the noise in them. Also, the assignation of colours according to the orientation is coherent through all the input images considered.

3 Segment Detection

As previously mentioned, the second stage of our framework is the line segment detection in the input image through an efficient combination of the fuzzy Hough transform and the Burns segment detector. A preliminary version of such segment detector, very accurate but also inefficient, is described in [11]. This section contains a brief introduction to the fuzzy Hough transform and the Burns segment detector and a description of how these two algorithms can be efficiently combined.

The fuzzy Hough transform is widely used in Computer Vision for the detection of geometric shapes that can be defined through parametric equations. When applied to the detection of 2D lines in digital images, the operation of the Hough transform is based on the normal equation of the line ($\rho = x \cdot \cos\theta + y \cdot \sin\theta$). Concretely, the continuous (ρ, θ) space is quantised into rectangles of size $(\triangle\rho \times \triangle\theta)$ and each rectangle is assigned to a position in an bidimensional accumulator array A.

During the operation of the algorithm, each input pixel $P = (x, y)$, with its corresponding (ρ_P, θ_P), votes for a set of cells in A. The contribution to a cell $A(\rho, \theta)$ is proportional to the distance between $(\rho - \rho_P)$ and (θ, θ_P). Once the voting process has finished, a maxima detection process takes place, each maximum in the accumulator array corresponds to a line in the input image that can be composed of one or several segments.

Although the fuzzy Hough is a very popular line segment detector, it has several important disadvantages. First, the number of pixels that can vote for a line depends on its position (i. e., the process is biased against the lines that lie close to a corner, favouring the diagonals or the lines parallel to the boundaries).

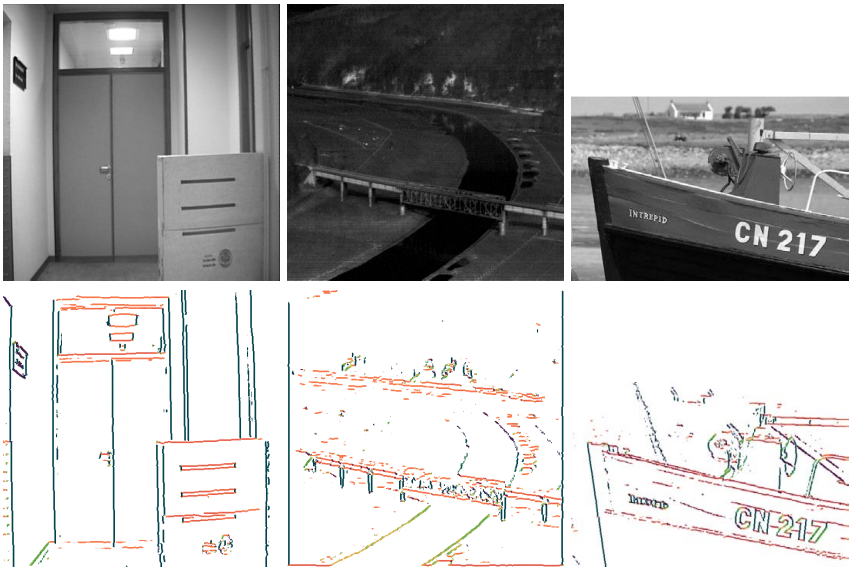


Fig. 2. First row: input images. Second row: low-level directional primitive detection results.

Also, since the Hough transform is a global process, the information in the whole image can motivate the deviation of some line segments and the detection of spurious segments [11]. Due to these disadvantages, we have combined the fuzzy Hough transform with some principles from the Burns segment detector.

As opposed to the fuzzy Hough transform, the Burns segment detector operates locally on intensity images by organising the pixels into line-support regions according to their orientation and extracting the segments in each line-support region separately from the underlying intensity information.

In order to group the input pixels into line-support regions, the π rad orientation range is divided into 4 partitions starting at 0 rad, each pixel is assigned a label in these partitions according to its orientation and position, and the simple connected-components algorithm is used to determine the line-support regions. If the orientation estimation is correct, all the pixels in a line segment are assigned to the same partition. When the orientation of a line segment lies close to a partition boundary, its pixels can be assigned to both adjacent partitions and the line segment can be fragmented. In order to solve this fragmentation problem, a second set of partitions, that divides the π rad orientation range in 4 partitions starting at $\frac{\pi}{8}$, is introduced. In order to merge both sets of partitions, each pixel is assigned to the largest line-support region possible. At the end of this process, each line-support is identified through a unique label.

In a preliminary implementation of the line segment detector proposed [11], the low-level directional primitives were first organised into line support regions and the fuzzy Hough transform was then applied to each line-support region separately. Although this implementation produced impressive results, it had an

important drawback that seriously limited its applicability, the high computational complexity, since an image containing N line support regions involved the computation of N independent fuzzy Hough transforms.

The next subsection introduces an efficient combination of the fuzzy Hough transform and the Burns segment detector that dramatically reduces the complexity of the previous implementation while preserving its good results.

3.1 Efficient Combination of the Fuzzy Hough Transform and the Burns Segment Detector

As previously mentioned, the fuzzy Hough transform and the Burns segment detector are two diametrically different algorithms that, combined, produce very impressive results. While the fuzzy Hough transform is based on the results of a low-level feature extraction process, the Burns segment detector operates directly on the intensity images. Also, while the operation of the fuzzy Hough transform is global and based on a bidimensional accumulator array $A(\rho, \theta)$, the Burns segment detector organises the input pixels into line-support regions and detects the line segments in each region separately.

In order to efficiently combine these two algorithms into a line segment detector capable of operating on our pseudo-colour input images (see sec. 2) the operation of the fuzzy Hough transform has been discretised through the addition of a third dimension *label* to the accumulator array A, in such a way that each cell $A(\rho, \theta, label)$ represents a line in the line-support region *label* instead of a line in the whole image.

The algorithm starts by grouping the input edge pixels into line-support regions identified by unique labels, as described in sec. 3. Each input pixel $P = (x, y)$ is now identified by its corresponding ρ_P, θ_P and $label_P$. During the voting process, the contribution of P to the cells $A(\rho, \theta, label)$ is inversely proportional to the distances $d(\rho_P, \rho)$ and $d(\theta_P, \theta)$ only if $(label_P = label)$, since the voting space is now locally organised into disjoint line-support regions.

Fig. 3 shows the results of the line segment detector proposed when applied to the images in the first row of fig. 2. As fig. 3 shows, the segments in the underlying images are very accurately detected from the low-level directional primitives.

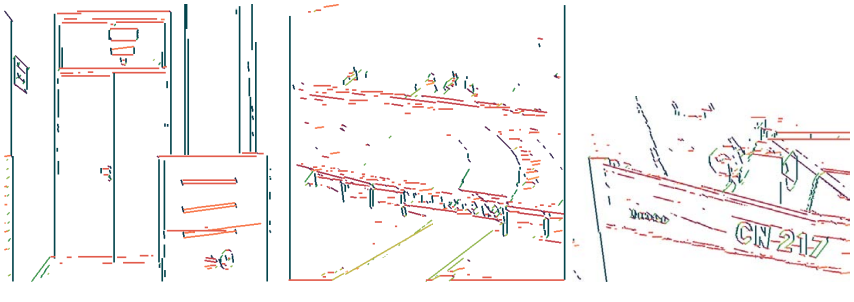


Fig. 3. Results of the segment detector on the images in the first row of fig. 2

The computational complexity of the line segment detector proposed is similar to that of the fuzzy Hough transform, as table 1 demonstrates. Table 1 shows the computational time of the implementation just described when applied to the images in the first row of fig. 2 and demonstrates that the computational time of the proposed implementation is very similar to that of the fuzzy Hough transform and extremely lower than that of the implementation described in [11].

Table 1. Computational time of the fuzzy Hough transform, the previous implementation and the proposed implementation of the line segment detector

	Hough	Hough+Burns prev	Hough+Burns def
boat	3 sec	59 sec	8 sec
corridor	3 sec	26 sec	5 sec
bridge	3 sec	60 sec	8 sec

Also, since the variable *label* can take arbitrarily large values and the accumulator array A is very sparse, in order to minimise the memory requirements of the proposed implementation, A has been implemented as a 2D static array of dynamic lists implemented as hash tables [5], instead of as an static 3D array.

4 Conclusions

This paper describes a computational framework for line segment detection in 2D images. The first stage of the framework is the low-level primitive extraction and organisation through the Gabor decomposition and a growing cell structure, respectively. The second stage is the line segment detection, implemented through an efficient combination of the fuzzy Hough transform and the Burns segment detector.

These two line segment detectors, very popular in the literature, have some important drawbacks: the fuzzy Hough transform operates globally on the image, which produces good results when applied to the detection of complex shapes like circles or ellipses, but causes poor results for line segment detection [11]; also, the Burns segment detector operates directly on the intensity image, which difficults its combination with the results of low-level feature detection algorithms.

The main contribution of this paper is the description of a line segment detector that efficiently combines principles from the fuzzy Hough transform and the Burns segment detector. The line segment detector proposed outperforms the fuzzy Hough transform [11] at a low computational cost, as table 1 shows. Also, as opposed to the Burns segment detector, that operates on intensity images, our line segment detector can be applied to the result of any low-level feature detector that provides information about the feature orientation like, for example, the Sobel edge detector.

The line segment detector introduced in this paper has been tested over a wide range of input images. Some of these images can be accessed at the url http://www.lfcia.org/staff/marta/EUROCAST2007.

Acknowledgements

This work has been partly supported by the University of A Coruña and the Xunta de Galicia through grant PGIDIT06TIC10502PR and by the Dirección Xeral de I+D.

References

1. Burns, J.B., Hanson, A.R., Riseman, E.M.: Extracting straight lines. IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8(4), 425–455 (1986)
2. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Analysis and Machine Intelligence 8, 679–698 (1986)
3. Van Deemter, J.H., Du Buf, J.M.H.: Simultaneous detection of lines and edges using compound Gabor filters. International Journal of Pattern Recognition and Artificial Intelligence 14(4), 757–777 (2000)
4. Fritzke, B.: Growing cell structures - a self-organizing network for unsupervised and supervised learning. Neural Networks 7(9), 1441–1460 (1993)
5. Fung, P.-F., Lee, W.-S., King, I.: Randomized generalized hough transform for 2-d grayscale object detection. In: Proc. of the 13th International Conference on Pattern Recognition, pp. 511–515 (1996)
6. Gabor, B.: Theory of communication. Journal of the Institute of Electronic Engineers 36(93), 429–457 (1946)
7. Hough, P.V.C.: Method and means for recognizing complex pattern. U. S. Pattern 06954 (1962)
8. Lindeberg, T.: Edge detection and ridge detection with automatic scale selection. International Journal of Computer Vision 30(2), 117–154 (1998)
9. Nalwa, V.S., Binford, T.O.: On detecting edges. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8(6), 699–714 (1986)
10. Penas, M., Carreira, M.J., Penedo, M.G.: Gabor wavelets and auto-organised structures for directional primitive extraction. In: Perales, F.J., Campilho, A., Pérez, N., Sanfeliu, A. (eds.) IbPRIA 2003. LNCS, vol. 2652, pp. 722–732. Springer, Heidelberg (2003)
11. Penas, M., Carreira, M.J., Penedo, M.G., Mariño, C.: Segment detection using burns principles in a pseudocolor hough transform. In: Marques, J.S., Pérez de la Blanca, N., Pina, P. (eds.) IbPRIA 2005. LNCS, vol. 3523, pp. 175–182. Springer, Heidelberg (2005)
12. Sarkar, S., Boyer, K.L.: Optimal finite impulse response zero crossing based edge detectors. CGVIP: Image Understanding 54, 224–243 (1991)
13. Wyszecki, G., Stiles, W.S.: Color science, concept and methods, quantitative data and formulae. John Wiley & sons, Chichester (1982)
14. Zheng, Y., Li, H., Doermann, D.: A model-based line detection algorithm in documents. In: Proc. of the 7th International Conference on Document Analysis and Recognition (2003)

Using Fisher Kernel on 2D-Shape Identification

Carlos M. Travieso[1], Juan C. Briceño[2], Miguel A. Ferrer[1], and Jesús B. Alonso[1]

[1] CEntro Tecnológico para la Innovación en Comunicaciones (CETIC)
Dpto. de Señales y Comunicaciones, Universidad de Las Palmas de Gran Canaria
Campus de Tafira, E35017, Las Palmas de Gran Canaria, Spain
{ctravieso,mferrer,jalonso}@dsc.ulpgc.es
[2] Escuela de Ciencias de la Computación e Informática. Universidad de Costa Rica.
Sede "Rodrigo Facio Brenes", Montes de Oca, 2060, San José, Costa Rica
briceño@ecci.ucr.ac.cr

Abstract. This paper proposes to use the Fisher kernel for planar shape recognition. A synthetic experiment with artificial shapes has been built. The difference among shapes is the number of vertexes, links between vertexes, size and rotation. The 2D-shapes are parameterized with sweeping angles in order to obtain scale and rotation invariance. A Hidden Markov Model is used to obtain the Fisher score which feeds the Support Vector Machine based classifier. Noise has been added to the shapes in order to check the robustness of the system against noise. Hit ratio score over 99%, has been obtained, which shows the ability of the Fisher kernel tool for planar shape recognition.

Keywords: 2D-shape recognition, 2D-shape, Hidden Markov Models (HMM), Support Vector Machines (SVM), Fisher Kernel.

1 Introduction

Object recognition, shape analysis, and classification constitute important research areas in computer vision. A large number of approaches for three-dimensional object recognition are based on the analysis of two dimensional (2D) aspects of objects [1].

Nowadays, the advances produced in computers and learning algorithms have permitted us to work each time, on more applications. Particularly, the 2D pattern recognition is a very extended and an important field due to their importance in different types of areas, as Medicine, OCR applications, Biology, etc.

In this context, the aim of this paper is to propose a hybrid Hidden Markov Model [2] (HMM) / Support Vector Machine [3] (SVM) for 2D shape classification by Fisher kernel.

Hidden Markov Models [2] are a widespread approach to probabilistic sequence modelling and have been extensively employed in several applications in the last decade. The use of HMMs for shape analysis has already been researched (see [4][5] for example) with quite interesting results. However, it is well known that the problem of the HMM classifier is its limited discriminative power; therefore, we have used another classifier. SVMs [6] have been recognized as powerful tools for the task of classification with a good discriminative ability, although inefficient for modelling temporal variations.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 740–746, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In order to get the complementary advantages of both, hybrid HMM and SVM structures have been proposed. These approaches map the input spatial-temporal sequence to a fixed dimensional vector in the so called score-space by means of the HMM using Fisher kernel [7]. Then the SVM is applied to the score space vector to recognize the input sequence.

In this paper, we propose the use of Fisher kernel from HMM and the classification of that score by SVM classifier, for recognizing planar objects and report its performance in the cases of different disturbances: intra/inter class variability using noise with a uniform probability density function. It have been compared with the HMM classifier. Mostly, this work is the previous step to the use of these algorithms to real applications, because in them, shapes always are different between them. In the next figure, it can be observed the block diagram of system proposed.

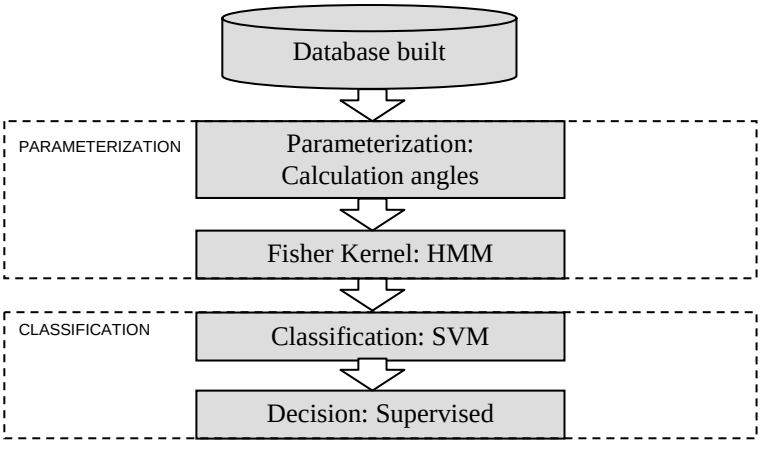


Fig. 1. Block diagram of system proposed

2 Database Built

The used methodology is based on working with binary images; a database of 21 classes has been designed from geometric figures, such as polygons with several sides, stars of several ends, etc.

In our database, 100 samples per each class have been generated, having 2100 images. To these figures have been introduced a random noise, with a uniform probability density function, and it have built 100 different figures for each one of the classes. In the following figure, some samples can be observed. Also, we can see the difference inter-class with respect to size and rotation.

3 Parameterization

These images have been parameterized, and the goal in this process is to pass from two-dimensional shape to a vector (one dimension), for it has swept the shape with

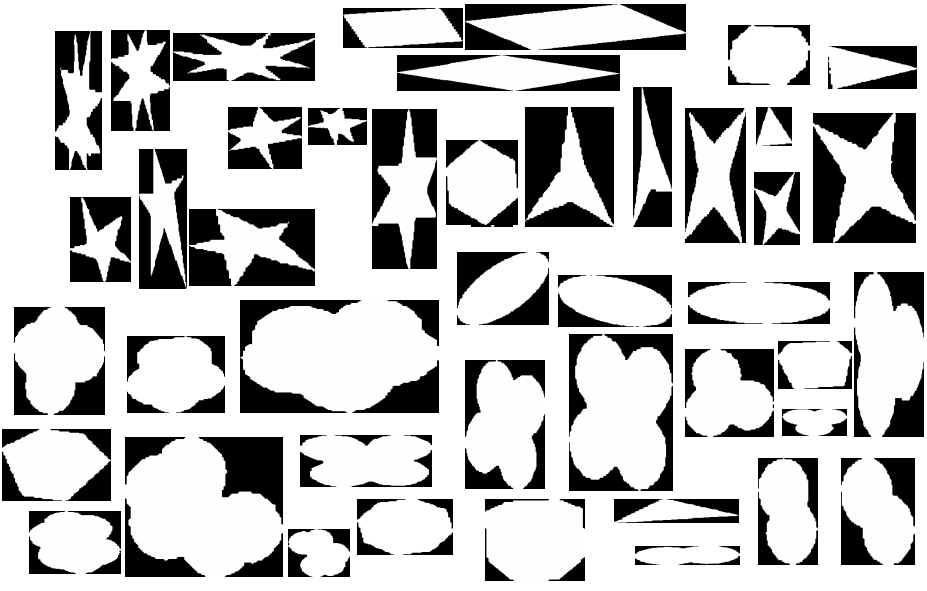


Fig. 1. Example of an angular coding for a 30 control points selection

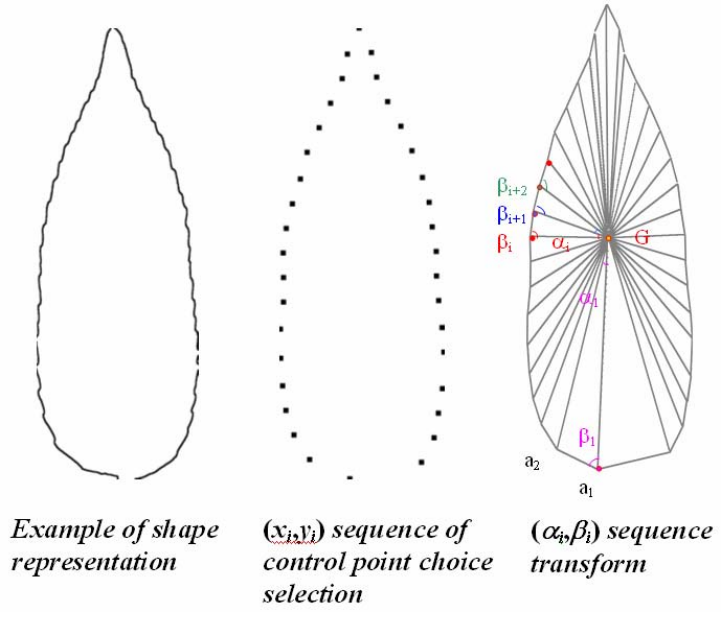


Fig. 2. Example of an angular coding for a 30 control points selection

polar coordinates from its centroide to exterior contour. The initial point is on the zero degrees direction. We have made a parameterization, which has been carried out for

obtaining rotation, translation and scale invariance. It is based from angles (α and β), which are shown in the figure 2. Where, α is angle between radiuses from centroide, and β is angle between the radius from centroide and the planar shape.

These parameters are calculated for a number of contour points, where that number of points is an input variable of our algorithm. In particular for this work, we have fixed this number on 360, each point for one degree of the shape. In the figure 2, it can be seen an example with 30 points, and its sweeping.

The way, to choose the number of sweeping points, has been carried out for establishing the minimum error after of its reconstruction, for which are going to choose the most significant points of the contour, therefore, the chosen points are the most representative of the shape. More information about this method can be observed on [8].

The following step is its automatic recognition. These values are taken as correlative points, and for this reason, they can be considered as a sequence. So, the classifier can be chosen for working with sequence.

4 Classification Task: Fisher Kernel

Hidden Markov Models (HMM) [1] is a good known classifier for its application on spatial-temporal sequences, as for example, on voice applications [9], where it has been widely used. An HMM [2] models a temporal sequence by means of a finite state machine. When an input sequence O of length T feeds an HMM model denoted by λ, the sequence travel through the HMM states governed by the state transition probability and the observation symbol probability emitting the probability $P(O|\lambda)$ of the sequence belongs to the HMM.

The HMM that models each class is obtained by means of the Baum-Welch algorithm, and the $P(O|\lambda)$ values are obtained by means of the forward-backward procedure. The gamma matrix $\gamma_t(i)$ is defined as the probability of being in state *i* at time *t*.

Once trained the HMMs $\{\lambda_c\}_{1\le c\le C}$ that model each one of the C possible classes, the classification process assigns the unknown input sequence O to the class that maximizes $P(O|\lambda_c)$.

The score spaces is a fixed dimension space where variable length sequences are mapped. T Jaakkola et al [7] suggest as score space the so called Fisher kernel which is defined as:

$$U_O = \nabla_\theta \log P(O|\lambda) \tag{1}$$

where each component of U_O is the derivate with respect to a particular parameter of the HMM and it specifies the extent to which each parameter contributes to the input sequence.

El valor de dicha derivada especifica la extensión con que cada parámetro contribuye a la secuencia de entrada. Específicamente, Jaakkola et al utilizaron la

derivada respecto a la matriz de probabilidad de emisión de símbolos $\left\{ b_j(v_k)\Big|_{1\le k\le M} \right\}_{1\le j\le N}$. From the derivate obtained by Jaakkola [7], we have obtained the matrix of Fisher score by the next expression, following with nomenclature of Rabiner for HMM; [2]:

$$U(i,k) = \frac{\partial}{\partial b_i(v_k)} \log P(O|\lambda) = \frac{\sum_{t=1}^{T} \gamma_t(i)\delta(\|v_k - O_t\|)}{b_i(v_k)} \quad 1 \le k \le M, 1 \le i \le N \tag{2}$$

being δ the Dirac delta function. Fisher adds a third term, which subtracted a certain value with the previous expression 2. That value has not been taken into account because it does not contribute information on the classification process. In expression 2, the numerator shows the number of times that we have used each symbol in each state. And the denominator shows the probability of use for each state. In the case of perfect overlapping between the model and the input sequence, all the *N*x*M* components of the matrix $U(i,k)$ will have the same value. If the variance of that matrix is big, then the difference between the model and the input sequence will be big.

Those matrixes, one per each group of parameters of each characteristic sequence, are the space, which will be used by SVM for the identification of the planar objects.

The SVM [6] is a binary classifier. Its principle relies on looking for a linear hyperplane that maximizes the distance (margin) between the separator hyperplane and the data. To construct the multi-class classifier we have trained one SVM per class taking the samples of its class as positives and the samples from all other classes as negative. An unknown new sample will be assigned to the class of the SVM with the largest output.

Therefore, the scheme works as follows: the input vector sequence that represents the unknown 2D shape is mapped into the score space by the C trained HMMs $\{\lambda_c\}_{1\le c\le C}$ obtaining C score space vectors of dimension N. The score space vectors are concatenated in a C by N dimension vector which will be classified by the multiclass SVM.

5 Experiments and Results

A supervised mode has been used by the classification system, and besides, results are shown with its average and typical deviation, because each experiment has been repeated five times. The percentage of samples used for training are from 15% to 30% of the database created, and for each experiment the samples have been chosen randomly, with a probability density function. The remainder of the samples, from 85% to 70%, they have used to carry out the test or evaluation of the system.

The results obtained in these experiments have not exceeded a success rate of 30% (see table 1), therefore, we have decided to employ a more efficient classifier. The goal is to use the previous work, and for this, our beginning point is the HMM classifier. The new parameters are obtained from HMM, in particular, from the forward and backward coefficients, bellowed to HMM. With these parameters, we have calculated the called Fisher kernel [7]. Now, we have parameters with

information about the creation of HMM with invariance parameters (α and β) from figures contour. To find values more discriminates, it has been made tests with various states in the HMM. For the remainder of HMM variables, we have been fixed their values with 16 symbols by states and use of multi-labelled [10]. Too, it has been used the Baum-Welch and Viterbi algorithm, for the classification problem. The software used is the GPDShmm toolbox [10].

This information has been classified with a Support Vector Machines (SVM) [6], because it is a robust classifier. We have worked with RBF kernel. SVM works with a binary classification; therefore, we have built a multi-class classifier, for each one of the 21 different classes. The technique implemented has been "one vs. remainder", and it has generated very satisfactory results, with an averaged success rate of 99.71% (see table 1). The software used is the SVMlight [11]. On table 2, we can see the results with SVM classifier, for the comparison between methods.

For these experiments, we have used the same methodology than previous experiments (supervised classification and each experiment repeated 5 times). Although, the supervised classification has changed, we have used from 5% of samples for training and 95% for the test, to 30% of the samples for training and the remainder for the test.

Table 1. Results with SVM using HMM with 50 states

% for training	Kernel Lineal	Kernel RBF	Value of g (RBF)
3	50.08% ±41.78	50.14% ±40.81	3×10^{-7}
4	89.95% ±2.83	87.23% ±4.09	2×10^{-7}
7	94.77% ±1.39	92.31% ±0.68	2×10^{-7}
11	96.77% ±1.62	95.25% ±1.62	3×10^{-7}
14	98.23% ±0.34	97.18% ±0.29	2×10^{-7}
18	98.89% ±0.46	98.38% ±0.78	2×10^{-7}
21	99.48% ±0.13	99.05% ±0.28	2×10^{-7}
25	99.58% ±0.27	99.39% ± 0.39	1×10^{-7}
30	99.71% ±0.14	99.59% ± 0.17	3×10^{-7}

Table 2. Results with SVM

	Kernel RBF	Value of g (RBF)
15	39.11% ± 1.42	2×10^{-2}
20	40.08% ± 1.18	1×10^{-2}
25	42.55% ± 0.69	1×10^{-2}
30	43.35% ± 1.54	1×10^{-2}

6 Conclusion

A very robust system has been created, which is strong versus the noise for the automatic detection of 2D-shape. For this, we have used the Fisher kernel, generated from HMM, and classified with SVM. The averaged success rate has been 99.71%. Seeing the table 1 and using 20% from our database for training, results are upper than 99%.

The robustness of this HMM/SVM (Fisher Kernel) on noise (variability of size and rotation) has been tested. The hybrid system has proved to be more robust than the SVM to that kind of alteration. These results, along with the others published in existing literature, confirm the HMM as a powerful approach to 2D shape classification, and emphasise the ability of Fisher kernel based systems to improve the HMM performance.

References

1. Loncaric, S.: A Survey of Shape Analysis Techniques. Pattern Recognition Nº(8), 983–1001 (1998)
2. Rabiner, L.R.: A tutorial on Hidden Markov models and Selected Applications in Speech Recognition. Proceedings of the IEEE 77(Nº2), 257–286 (1989)
3. Müller, K.R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An Introduction to Kernel-Based Learning Algorithms. IEEE Transactions on Neural Networks 12(Nº 2), 181–201 (2001)
4. Cai, J., Liu, Z.Q.: Hidden Markov Models with Spectral Features for 2D Shape Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (Nº12), 1454–1458 (2001)
5. Bicego, M., Murino, V.: Investigating Hidden Markov Models Capabilities in 2D Shape Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(Nº 2), 281–286 (2004)
6. Burges, C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery nº(2), 121–167 (1998)
7. Jaakkola, T., Diekhans, M., Haussler, D.: A discriminative framework for detecting remote protein homologies. Journal of Computational Biology Nº(1-2), 95–114 (2000)
8. Briceño, J.C., Travieso, C.M., Ferrer, M.A., Alonso, J.B., Briceño R.D.: A genus recognition system for the Costa Rica Lauraceae Family, using a Support Vector Machine. XXIII Conferencia Latinoamericana de Informática, pp.120 (abstract) (2006)
9. Rabinier, L., Juang, B.H.: Fundamentals of Speech Recognition. Prentice-Hall, Englewood Cliffs, New Jersey (1993)
10. David, S., Ferrer, M.A., Travieso, C.M., Alonso, J.B.: gpdsHMM: A Hidden Markov Model Toolbox in the Matlab Environment. CSIMTA, Complex Systems Intelligence and Modern Technological Applications, 476-479 (2004)
11. Joachims, T.: Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge (1999)

Benefits of Plugin-Based Heuristic Optimization Software Systems

Stefan Wagner[1], Stephan Winkler[2], Erik Pitzer[2], Gabriel Kronberger[2], Andreas Beham[2], Roland Braune[2], and Michael Affenzeller[1]

Upper Austrian University of Applied Sciences
[1] Department of Software Engineering
[2] Research Center Hagenberg
Softwarepark 11, 4232 Hagenberg, Austria
{swagner,swinkler,epitzer,gkronber,abeham,
rbraune,maffenze}@heuristiclab.com

Abstract. Plugin-based software systems are the next step of evolution in application development. By supporting fine grained modularity not only on the source code but also on the post-compilation level, plugin frameworks help to handle complexity, simplify application configuration and deployment, and enable users or third parties to easily enhance existing applications with self-developed modules without having access to the whole source code.

In spite of these benefits, plugin-based software systems are seldom found in the area of heuristic optimization. Some reasons for this drawback are discussed, several benefits of a plugin-based heuristic optimization software system are highlighted and some ideas are shown, how a heuristic optimization meta-model as the basis of a thorough plugin infrastructure for heuristic optimization could be defined.

1 Introduction

In software engineering a high degree of modularity is generally considered to be an import software quality aspect ([6], [2]). Having to deal with changing requirements, new feature requests and short development times is the daily life of an application developer. So trying to build software systems that are divided into clearly separated modules communicating over well-defined interfaces is a critical factor of success. On the one hand, highly modular software systems encourage code reuse as several generic modules could be used in other projects. On the other hand, reacting on change and extension requests is easier as only the directly involved parts have to be replaced without needing to change (or even know) the application as a whole.

So when we take a look at the evolution of software development methodologies the trend towards more modularity is obvious. Starting with monolithic programs, going through first procedural and then object-oriented programming and ending with component-oriented or aspect-oriented development of complex software systems, one can see that in each step of development new concepts

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 747–754, 2007.
© Springer-Verlag Berlin Heidelberg 2007

were introduced to support clearer structuring and separation of software into modules (procedures, objects, components, aspects, etc.). However, all these decomposition approaches are mostly applied at the source code level. When it comes to compilation, the different modules of an application are often still linked to a single monolithic executable. When changing a module or providing an additional extension or new functionality is required, the changes in source code are clearly separated but the whole application has to be re-compiled or at least re-linked. This fact makes it hard to dynamically extend software systems on demand (especially with new modules that are for example developed by third parties).

As a consequence, plugin-based software systems became very popular, as they are able to overcome this problem. By not only splitting the source code into different modules but compiling these modules into enclosed ready to use software building blocks, the development of a whole application or complex software system is reduced to the task of selecting, combining and distributing the appropriate modules. Due to the support of dynamic location and loading techniques offered in modern application frameworks as for example Java or .NET, the modules do not need to be statically linked during compilation but can be dynamically loaded at runtime. Usually, a specific plugin management mechanism takes care of this task by finding all available parts, loading them on demand and providing mutual interaction by creating objects of specific types implementing required interfaces (contracts).

In the following a short overview of several existing plugin frameworks for different platforms is given (see Section 2). After pointing out general benefits of plugin-based application development in Section 3, the main focus lies on discussing plugin approaches in the light of heuristic optimization software systems (Section 4), as the authors think that especially in this area plugin-based approaches should be very effective but are not implemented in most of existing heuristic optimization frameworks (at least in those known to the authors). Finally a summary and concluding remarks are stated in Section 5.

2 Existing Plugin Frameworks

Especially in Java application development the idea of enhancing software with a plugin mechanism is more and more frequently realized. The OSGi Alliance offers a well developed and standardized architecture for plugin-based and service-oriented software systems called the OSGi Service Platform Specification [8]. The OSGi architecture is structured in several layers built on top of the Java Runtime Environment and providing mechanisms (enhanced class loading, module management, communication and cooperation between modules) required for application building. The application itself (also called bundle) can be situated on any layer depending on the required functionality. The layout of the OSGi architecture is shown graphically in Figure 1. Successful implementations of the OSGi Service Platform Specification are documented in several publications (for example [3]) as well as in open source and commercial software projects.

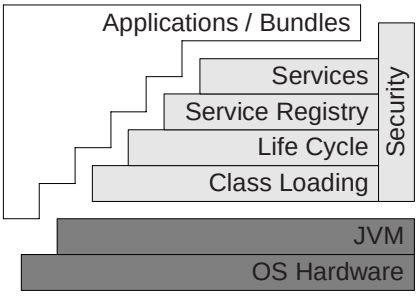


Fig. 1. OSGi Architecture (taken from [7])

Beside the OSGi Service Platform, also the well known Eclipse development platform and in particular the Eclipse Rich Client Platform (Eclipse RCP) intensively use the concept of plugins ([1], [12], [5]). Additionally, other popular applications can be found that follow similar ideas like the Mozilla web browser [11] or the NetBeans environment.

However, the idea of plugin oriented application development is not restricted to the Java platform at all. Realizing a powerful and flexible plugin infrastructure is possible with any kind of application development environment offering basic features such as dynamic loading and reflection. Several publications show how to implement plugin-based applications using the .NET Common Language Runtime ([9], [10], [4], [15]) and underpin the worth of this approach. Furthermore, the Razor project of Mark Belles also has to be mentioned, which goes beyond scientific analysis and prototypical implementation and focuses on developing a fully functional plugin framework based on the .NET framework.

3 General Benefits of Plugin-Based Application Development

As already stated in the introduction, a high degree of modularity is an important factor for building high quality software systems. Apart from the typical improvements of testability, reusability, readability, and maintainability that are already supported by classical modularization techniques such as object-oriented modeling, plugin-based software development further simplifies the following aspects discussed in the next subsections:

Complexity

Thinking of modularity not only at the source code level but also on the application level helps to handle the high degree of complexity large software systems have to deal with today. By separating an application into several plugins, where each plugin is responsible for a well-defined task, the complexity and size of a single module is drastically reduced. Furthermore, when compiling a plugin into

an executable unit, the plugin is effectively sealed and can be shared among different developers without having the need (or even the chance) to take a look at the original source code. As communication and interaction with the plugin is done using fixed interfaces, a developer does not have to be bothered with trying to understand other developers' source codes. Furthermore, the development of drivers and stubs to test each module independently is also simplified. However, to be honest complexity does not vanish. Instead it is transferred from the programming level up to the modeling level, as the final application has to be assembled by combining different small and simple plugins. Thereby the problem of glue code arises that is required to link and combine these modules. How to handle incompatibilities between different plugins and how to prevent the complexity from being shifted from the application modules into the glue code holding the whole application together is still an active field of research. For example, at this point generative software development comes into play as a concept for automatically generating glue code from system models instead of manually putting the required parts together.

Configuration, Customization, Deployment

In plugin-based software systems the main application's logic is reduced to the mere plugin hosting and management mechanism and does not offer any application specific functionality. The set of features of such an application is defined by different plugins loaded and managed by the plugin infrastructure. So it is much easier to customize a plugin-based software system to the specific requirements in a customer's scenario just by adding necessary and removing irrelevant parts. Besides, application deployment is simplified as functionality to deploy new or to update outdated plugins from update locations (i.e. servers) over a network can be realized easily.

Expendability

Finally, also users benefit from a plugin-based infrastructure, as they are equipped with a comprehensive and yet easy to use application programmer's interface (API). Extending an application with new individual features without understanding or even having access to the full source code of an application is plain sailing (just implement new plugins and register them in the application). This also offers the possibility for third parties to develop and sell extensions to an existing application independently.

4 Plugin-Based Heuristic Optimization Software Systems

Although the benefits of plugin-based software systems are well known in the software engineering community and several open source and commercial projects (especially related to development environments) demonstrate its use, in the area of

heuristic optimization software systems such approaches are not commonly used [13] yet. So what is the reason that developers of heuristic algorithms lag behind recent improvements of software engineering?

In the opinion of the authors there are several answers to this question. On the one hand it is a matter of fact that most heuristic optimization experts are originating from other research areas (mathematics, mechatronics, economy, logistics, etc.) and are therefore not so familiar with advanced application development techniques in contrast to software engineers. However, it should be quite easy to overcome this drawback by (prototypically) implementing a plugin-based heuristic optimization software system and showing its benefits. For example, the HeuristicLab optimization environment [14] tries to fulfill this demand. More details can be found on the HeuristicLab homepage[1].

On the other hand the more severe reason is the lack of a common model unifying heuristic optimization algorithms. Such a heuristic meta-model is required as the basis for developing a practicable plugin infrastructure for this domain. However, due to the vast amount of different algorithm variants as for example Evolutionary Algorithms (Genetic Algorithms, Genetic Programming, Evolution Strategies, or Evolutionary Programming), other population-based algorithms (Ant Colony Optimization, Particle Swarm Optimization, e.g.) or neighborhood-based optimization techniques (Simulated Annealing, Tabu Search, etc.) this is not a straightforward task.

So let us consider all those different kinds of optimization techniques. What do all of them have in common? In fact, heuristic optimization is always about manipulating promising solution candidates in order to reach high quality regions of the search space in reasonable time. This process can be split into a few main operations:

- Create one or more initial solution candidates (using a specific solution representation)
- Evaluate solutions using a given quality measure
- Manipulate a solution
- Create new solutions out of existing ones
- Regroup solutions

So we have three different aspects interacting with each other: the solution encoding, the evaluation function and a set of operators working on single solutions or sets of solutions. Standardizing this interaction can be done by defining a small set of fixed interfaces: a solution can be abstracted to an object exposing one (single objective optimization) ore more (multiobjective optimization) quality values and containing data in the form of an instance of a solution representation (for example a multidimensional vector of real values, a permutation, or a tree-like data structure). A solution processing operator is simply a method taking a single solution (for example evaluation) or a set of solutions (for example selection) and returning a single solution or multiple solutions again. One possible modeling of these interfaces is shown in Figure 2.

[1] `http://www.heuristiclab.com`

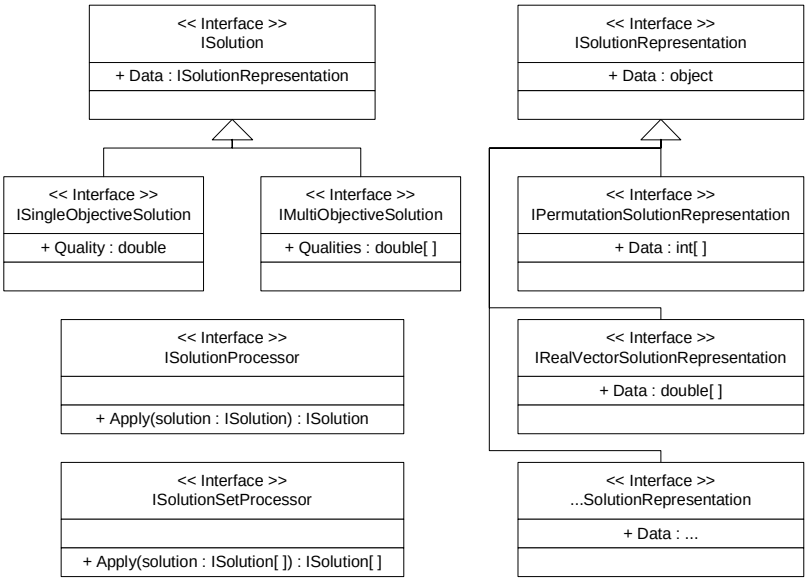


Fig. 2. Basic Heuristic Optimization Interfaces

Based upon this abstraction of heuristic optimization, a plugin-based approach can be realized. Interaction between the plugins is restricted to processing objects represented by the given interfaces leading to a high degree of exchangeability. A plugin itself is only responsible for a single and simple processing operation. When developing heuristic optimization software systems, additional benefits arise from applying this paradigm that go beyond the general aspects already discussed in Section 3.

Algorithm Development

By providing a set of plugins, each of which realizes a specific solution representation or operation, the process of developing new heuristic algorithms is revolutionized. Algorithms do not need to be programmed anymore but can be created by combining different plugins. This approach has a huge advantage: By providing a graphical user interface for combining plugins, no programming or software engineering skills are necessary for this process at all. As a consequence, algorithms can be modified, tuned or developed by experts of different fields (logistics, production, finance, or data analysis, e.g.) with less or no knowledge in the field of application development. This transfers development from software engineering to the concrete application domains profiting from the profound knowledge of the users and eliminating the communication overhead between optimization users and software engineers.

Experimental Evaluation

Secondly, the cumbersome and time-consuming task of evaluating different algorithmic approaches for a given optimization problem is simplified. New ideas can be prototyped rapidly (by simply adding new operators) and immediately evaluated within the application by comparing their results with existing classical algorithm configurations.

Combination of Different Optimization Paradigms

Finally, a plugin-based heuristic optimization software system helps to bridge the gap between different optimization paradigms like evolutionary algorithms, local neighborhood-based algorithms, swarm-based algorithms, or even exact mathematical optimization methods in general. Different concepts can be easily combined just by linking the relevant plugins. So for example realizing genetic algorithms equipped with a local optimization of solution candidates (memetic algorithms) is quite a walk-over. This will lead to algorithms that are individually tuned for specific applications bringing out the best of each paradigm.

5 Conclusion and Future Perspectives

In this paper the authors have highlighted the importance of modularity in complex software systems and discussed plugin-based application development as the next evolutionary step of programming paradigms. Several examples of plugin frameworks for modern development platforms such as Java or .NET show the benefits of this approach. These benefits are a better handling of complexity, a simplified configuration and deployment procedure and the ability to extend existing applications with new own or third-party modules without requiring access to the whole source code of an application.

However, when it comes to heuristic optimization software systems, modularity is mainly realized on the source code level alone by using classical decomposition techniques such as object-oriented modeling. Plugin-based approaches are seldom implemented. In the opinion of the authors this is a severe shortcoming of many systems in this domain. Having a plugin-based heuristic software system at hand would revolutionize the process of algorithm development as new heuristic algorithms do not have to be programmed anymore but can be modeled and assembled using a graphical user interface. As a consequence, no programming skills are required anymore, opening the field of algorithm engineering for many other users originating from different research areas such as logistics, production, finance, medicine, biology, or data analysis.

As one reason for the lack of plugin-based heuristic optimization software systems, the authors have identified the difficult task of developing a common meta-model for heuristic algorithms which is required as a basis for developing a practical plugin infrastructure. Some ideas how such a highly abstract meta-model might be defined are given in Section 4 and a prototypical implementation is available in form of the HeuristicLab optimization environment [14].

Future directions of research will focus on the one hand on further development of the plugin concept of the HeuristicLab environment. On the other hand a comprehensive set of heuristic optimization plugins would pave the way for realizing automation concepts for heuristic algorithm development. For Example, using Genetic Programming techniques to automatically evolve whole heuristic algorithms (i.e. different plugin combinations) in order to build new high-quality optimization algorithms specifically tuned to a given application is a very fascinating idea. Furthermore, fighting the problem of increased runtime (due to the overhead of complex plugin interactions) by using generative programming approaches is another research direction the authors will continue to work on.

References

1. Beck, K., Gamma, E.: Contributing to Eclipse. Addison-Wesley, London, UK (2003)
2. Cox, B.: Planning the software industrial revolution. IEEE Software 7(6) (1990)
3. Hall, R.S., Cervantes, H.: An OSGi Implementation and Experience Report. In: Consumer Communications and Networking Conference (2004)
4. Holm, C., Krüger, M., Spuida, B.: Dissecting a C# Application – Inside SharpDevelop. Apress (2003)
5. McAffer, J., Lemieux, J.-M.: Eclipse Rich Client Platform: Designing, Coding, and Packing Java Applications. Addison-Wesley, London, UK (2005)
6. McIllroy, M.: Mass produced software components. In: McIllroy, M. (ed.) Proceedings of the Nato Software Engineering Conference, pp. 138–155 (1968)
7. OSGi Alliance: About the OSGi Service Platform. Technical Report, OSGi Alliance (2005), `http://www.osgi.org`
8. OSGi Alliance: OSGi Service Platform Specification (Release 4). Technical Report, OSGi Alliance (2006), `http://www.osgi.org`
9. Osherove, R.: Creating a Plug-in Framework. Technical Report, Microsoft Developer Network (2003)
10. Osherove, R.: Search Dynamically for Plug-ins. Technical Report, Microsoft Developer Network (2003)
11. Shaver, M., Ang, M.: Inside the Lizard: A look at the Mozilla Technology and Architecture. Technical Report (2000), `http://www.mozilla.org`
12. Shaver, M., Ang, M.: Eclipse Platform Technical Overview. Technical Report, Object Technology International (2003), `http://www.eclipse.org`
13. Voss, S., Woodruff, D.: Optimization Software Class Libraries. Kluwer Academic Publishers, Dordrecht (2002)
14. Wagner, S., Affenzeller, M.: HeuristicLab: A Generic and Extensible Optimization Environment. In: Ribeiro, B., Albrecht, R.F., Dobnikar, A., Pearson, D.W., Steele, N.C. (eds.) Adaptive and Natural Computing Algorithms, pp. 538–541. Springer, Heidelberg (2005)
15. Wolfinger, R., Dhungana, D., Prähofer, H., Mössenböck, H.: A Component Plugin Architecture for the .net Platform. In: Lightfoot, D.E., Szyperski, C.A. (eds.) JMLC 2006. LNCS, vol. 4228, pp. 287–305. Springer, Heidelberg (2006)

Metaheuristic Approaches for Optimal Broadcasting Design in Metropolitan MANETs

E. Alba[1], A. Cervantes[2], J.A. Gómez[4], P. Isasi[2], M.D. Jaraíz[4], C. León[3], C. Luque[2], F. Luna[1,⋆], G. Miranda[3], A.J. Nebro[1], R. Pérez[2], and C. Segura[3]

[1] Computer Science Department, University of Málaga
[2] Computer Science Department, University Carlos III of Madrid
[3] Department of Statistics, O.R. and Computation, University of La Laguna
[4] Department of Informatics, University of Extremadura

Abstract. Mobile Ad-hoc Networks (MANETs) are composed of a set of communicating devices which are able to spontaneously interconnect without any pre-existing infrastructure. In such scenario, broadcasting becomes an operation of tremendous importance for the own existence and operation of the network. Optimizing a broadcasting strategy in MANETs is a multiobjective problem accounting for three goals: reaching as many stations as possible, minimizing the network utilization, and reducing the duration of the operation itself. This research, which has been developed within the OPLINK project (`http://oplink.lcc.uma.es`), faces a wide study about this problem in metropolitan MANETs with up to seven different advanced multiobjective metaheuristics. They all compute Pareto fronts of solutions which empower a human designer with the ability of choosing the preferred configuration for the network. The quality of these fronts is evaluated by using the hypervolume metric. The obtained results show that the SPEA2 algorithm is the most accurate metaheuristic for solving the broadcasting problem.

1 Introduction

With the rapid development of wireless communications technologies and the proliferation of mobile devices like cell phones, PDAs or laptops, mobile *ad hoc* networks (MANETs) have emerged as an important research field in current and future communication networks because they do not require infrastructure support and can be quickly deployed with low costs. MANETs consist of a collection of mobile, self-configurable hosts, called *nodes* or *devices*, which are free to move randomly and organize themselves arbitrarily. The mobility of devices along with the range-limited wireless links make the network topology to change rapidly and unpredictably over time. This dynamical behavior constitutes one of the main obstacles for performing efficient communications on such networks.

This work is focussed on the problem of broadcasting in MANETs. In such networks, broadcasting is not only the basic mechanism for solving many network

⋆ Corresponding author. Email: `flv@lcc.uma.es`

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 755–763, 2007.
© Springer-Verlag Berlin Heidelberg 2007

layer problems (e.g. routing) but also a common operation at the application level. Hence, choosing a broadcasting strategy properly will result in a major impact in network performance. In this paper, we are considering the problem of broadcasting over the so-called *Metropolitan* MANETs [1], where the density of devices is heterogeneous and continuously changing. This leads to networks composed of subsets of ad hoc networks that may merge and disjoin dynamically during the operation, so the network topologies change frequently (Fig. 1). In this context, rather than providing a generic protocol performing well on average situations, our proposal lies in optimally tuning the broadcasting service for a set of networks.

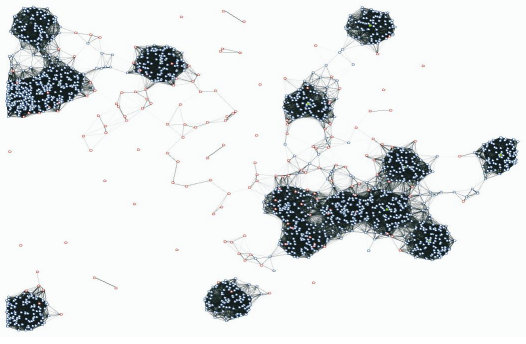


Fig. 1. An example of metropolitan MANET

Optimizing a broadcasting strategy is a multiobjective optimization problem (MOP), in which multiple functions have to be satisfied at the same time: maximizing the number of stations reached, minimizing the network use, and minimizing the duration of the process. In this work, the broadcasting strategy considered for optimization is DFCN [2], and the target networks are metropolitan MANETs. This MOP has been called DFCNT standing for *DFCN Tuning*. DFCNT is a real-world, complex optimization MOP which is very hard to be solved exactly within a reasonable amount of time. Heuristics become the choice here. These methods sacrifice the guarantee of finding the optimal solutions for the sake of (hopefully) getting accurate (also optimal) ones efficiently. Among them, metaheuristics [3] are well-known optimization algorithms that are based on combining basic heuristic methods in a higher level structured framework. The aim is to efficiently and effectively explore the search space of the given problem while seeking for its optimal solutions.

The goal of this work is to present a broad study in which seven different state-of-the-art multiobjective metaheuristics for solving the optimal broadcasting problem are evaluated. These metaheuristics are: NSGA-II [4] and SPEA2 [5] (genetic algorithms), MOPSO [6] (particle swarm optimization), AbYSS [7] (scatter search), cMOGA [8] (cellular genetic algorithm), and two adaptations of Evolution Strategies (ES) [9] and Differential Evolution (DE) [10] to the

multiobjective optimization field. We have been able to address this extensive study on such a complex problem thanks to the collaboration developed among the different research teams of the OPLINK project (`http://oplink.lcc.uma.es`). The optimizers have been compared by using the standard methodology in the multiobjective field and the drawn conclusions have been supported with an statistical analysis.

The paper is structured as follows. The next section provides the reader with a brief description of the DFCNT MOP that is being addressed in this work. Section 3 describes all the metaheuristics used. Metrics, parameterization, and results are presented in Sect. 4. Finally, conclusions and lines of future work are given in Sect. 5.

2 DFCNT: Problem Statement

Metropolitan mobile ad hoc networks are MANETs with some particular properties. First, they have one or more areas where the node density is higher than the average. They are called *high density areas*, and they can be statistically detected. Second, high density areas do not remain active full time, i.e., they can appear and disappear from the network. To deal with such kind of networks, there is no other solution than resorting to software simulations. In this work we have used *Madhoc* [11], a metropolitan MANET simulator (`www-lih.univ-lehavre.fr/~hogie/madhoc`). Very different realistic scenarios can be implemented within Madhoc (open areas, metropolitan environments, highways, etc.) because of its highly customizable behavioral parameters. We have configured the simulator for modelling a mall environment according to three main parameters: the size of the simulation area, the density of mobile stations, and the type of environment. For our experiments, we have used a simulation area of 40,000 square meters, a density of 2,000 devices per square kilometer, and a mall-like environment (see [8] for further details). We consider that the broadcasting is completed when either the coverage reaches 100% or it does not vary in a reasonable period of time (set to 1.5 seconds after some preliminary experimentation). This point is important since an improper termination condition will lead us to bad results or very slow simulations.

Once we have the target network, the broadcasting strategy used to be tuned is DFCN [2]. We have chosen this protocol because it has been specifically designed for metropolitan MANET scenarios. After carefully analyzing DFCN, we identified five parameters which determine the behavior of the protocol. For all the details about the DFCNT parameters, the reader is referred to [8]. These five parameters define five decision variables which correspond to a DFCN configuration. The objectives to optimize are: minimizing the duration of the broadcasting process, maximizing the network coverage, and minimizing the number of transmissions. Thus, we have defined a triple objective MOP, called DFCNT (*DFCN Tuning*). As we stated before, this problem is defined by a given target network in which the DFCN broadcasting strategy is used. We do not seek for a protocol performing well on every possible scenario.

3 Metaheuristics for Solving DFCNT

In this section we give a brief overview of the seven optimization methods used in this work. For further details, please use the given references on each algorithm.

- **NSGA-II:** The NSGA-II algorithm was proposed by Deb *et al.* [4]. It is a genetic algorithm based on obtaining a new population of individuals from the original one by applying the typical genetic operators (selection, crossover, and mutation); then, the most promising individuals for the next generation are chosen according to their rank and crowding distance. When it is used to solve continuous problems, NSGA-II applies binary tournament selection, SBX crossover and polynomial mutation.
- **SPEA2:** It is also a genetic algorithm, proposed by Zitler *et al.* in [5], in which each individual has assigned a fitness value that is the sum of its strength raw fitness and a density estimation. The algorithm applies the selection, crossover, and mutation operators to fill an archive of individuals; then, the non-dominated individuals of both the original population and the archive are copied into a new population. If the number of non-dominated individuals is greater than the population size, a truncation operator based on calculating the distances to the k-th nearest neighbor is used.
- **MOPSO:** This technique, proposed by Coello and Lechuga in [6], combines particle swarm optimization with the archiving strategy of PAES [12]. Non-dominated solutions in the swarm are stored in an external repository and they are used to guide the search. This repository is used both to store the solutions of the problem and to maintain a diverse population of particles. We have used the basic version of MOPSO [6], without the mutation operator.
- **AbYSS:** The Archive-Based hYbrid Scatter Search [7,13] is our own proposal to adapt the scatter search template to solve bounded continuous single objective optimization problems [14] to the multiobjective domain. The SBX crossover and the polynomial mutation operators are used in the solution combination and improvement methods, respectively. AbYSS uses an external archive for storing non-dominated solutions, which is managed by using the ranking and crowding of NSGA-II [4] as a niching measure.
- **cMOGA:** cMOGA [8] is a cellular genetic algorithm [15] which uses an external archive for storing the non-dominated solutions found during the search. The management strategy of this external archive is similar to the one used by PAES (as MOPSO does). This algorithm also uses SBX and polynomial mutation as genetic operators in its breeding loop.
- **Differential Evolution (DE):** We use an approach based on the reproduction scheme of DE [10] within the NSGA-II reproductive loop. That is, the algorithm works on a population of N vectors and, at each iteration, a new offspring population of size N is generated by using the DE scheme. The ranking and crowding mechanisms from NSGA-II are used to choose the N best vectors out of $2N$ for the next iteration.
- **Evolution Strategy (ES):** This multiobjective ES algorithm is a rather standard self-adaptive evolution strategy [9]; in particular, it is a $(\mu+\lambda)$ ES

using the selection scheme of NSGA-II (based on ranking and crowding distances). The mutation operator used is the gaussian mutation, which is applied using a variation of the 1/5 success rule [9].

4 Experiments

This section is devoted to present the experiments performed for this work. We firstly introduce the metric used for measuring the performance of the resulting Pareto fronts. Secondly, the parameterization of all the multiobjective algorithms is detailed. Finally, we thoroughly discuss and compare the results for DFCNT.

4.1 Quality Indicator: Hypervolume

We have used the Hypervolume [16] metric for assessing the performance of the multiobjective optimizers. This is a widely used, Pareto-compliant metric that measures both convergence and diversity in the resulting Pareto fronts. It is based on calculating the volume (in the objective space) covered by members of a non-dominated set of solutions Q. Let v_i be the volume enclosed by solution $i \in Q$. Then, a union of all hypercubes is found and its hypervolume (HV) is calculated:

$$HV = volume\left(\bigcup_{i=1}^{|Q|} v_i\right). \tag{1}$$

Algorithms with larger values of HV are desirable. Since this metric could be biased by arbitrary scaling of objectives, we have evaluated the metric by using normalized objective function values.

4.2 Parameterization

All the algorithms stop when 25,000 function evaluations of DFCNT have been computed. Due to the stochastic nature of Madhoc, five simulations per function evaluation have been performed, so that the fitness values of the functions are computed as the average resulting values of these five different trials. A maximum archive size of 100 non-dominated solutions is set in those algorithms using external archives. SBX crossover and polynomial mutation have been used in NSGA-II, SPEA2, AbYSS, and cMOGA. The detailed configuration of each algorithm is as follows:

- NSGA-II: It uses a distribution index of 10 for both SBX and polynomial mutation operators. The crossover and mutation rates have been set up to 0.9 and 0.2, respectively (as suggested by the designers of the algorithm).
- SPEA2: It has been configured with a population size of 100 individuals. The η_c value for SBX was fixed to 10 and the crossover rate to 1.0. In the polynomial mutation, η_m was set up to 20 and the mutation probability to 0.01.

- MOPSO: We have used 100 particles and 250 iterations to meet 25,000 function evaluations. The values for computing the new velocity vectors and new position of the particles are: $w = 0.4$, $c1 = c2 = 2.0$, and $X = 0.4$ (see [6] for the details).
- AbYSS: The configuration of both the SBX crossover and polynomial mutation (solution combination and improvement methods) uses a distribution index of 10. The size of the initial set P is 20, the number of iterations in the improvement method is 5, and the size of $RefSet_1$ and $RefSet_2$ is 10 solutions.
- cMOGA: It uses the same configuration for SBX and polynomial mutation as AbYSS. The algorithm has been configured with a population of 100 individuals arranged in a 10×10 square toroidal grid with a NEWs neighborhood, and using binary tournament selection, a crossover rate of 1.0, and mutation rate of 0.2.
- DE: From the set of predefined strategies, DE/rand/1/bin has been selected, which respectively means that a random vector is mutated, one single vector is considered for perturbation of this randomly chosen vector, and binomial crossover is used for recombination (see [10] for the details). Two additional parameters are needed to be assigned in DE, the weighting factor (F) and the crossover rate (CR). The chosen values for these parameters have been 0.5 and 1.0, respectively.
- ES: The ES algorithm has been configured with 100 individuals so that $\mu = \lambda = 100$ (and therefore 250 iterations). The step-size meta-control parameter Δ is equal to 0.7.

4.3 Results

We have made 30 independent runs of each experiment. The results are shown in Table 1 (best algorithms are at the top) which includes the median, $\tilde{x}$, and interquartile range, IQR, as measures of location (or central tendency) and statistical dispersion. Since we are dealing with stochastic algorithms and we want to provide the results with confidence, the following statistical analysis has been performed in all this work. First a Kolmogorov-Smirnov test is performed in order to check whether the values of the results follow a normal (gaussian) distribution or not. If so, the Levene test checks for the homogeneity of the variances. If samples have equal variance (positive Levene test), an ANOVA

Table 1. HV values

Algorithm	$\tilde{x}_{IQR}$
SPEA2	8.807e−01 $_{4.7e-03}$
ES	8.755e−01 $_{2.7e-03}$
NSGA-II	8.741e−01 $_{5.9e-03}$
AbYSS	8.687e−01 $_{3.8e-02}$
DE	8.672e−01 $_{8.4e-03}$
MOPSO	8.644e−01 $_{3.8e-03}$
cMOGA	8.480e−01 $_{5.4e-02}$

test is done; otherwise we perform a Welch test. For non-gaussian distributions, the non-parametric Kruskal-Wallis test is used to compare the medians of the algorithms. We consider here a confidence level of 95% (i.e., significance level of 5% or p-value under 0.05), which means that the differences are unlikely to have occurred by chance with a probability of 95%. After applying this test, we verify that statistical difference exists amongst the values included in Table 1.

From the values in the table, we can observe that SPEA2 obtains the best metric value, thus indicating that it is the most promising metaheuristic to solve DFCNT. This is an expected result since DFCNT is a three-objective MOP, and it is well-known that the density estimator of SPEA2 overcomes those used in NSGA-II and PAES (used in the rest of algorithms) in MOPs having more than two objectives [17]. In this sense, it is remarkable the HV values reached by the ES algorithm, which outperforms NSGA-II, the reference algorithm in multiobjective optimization. The last position of cMOGA in Table 1 is a consequence of its simplified archive management strategy, in which the individuals in the archive are not reused in the breeding loop of the algorithm. Anyway, the differences in the metric values are not large, indicating that all the algorithms compute similar fronts.

It is worth mentioning the time required to run each experiment. In a modern Intel Pentium 4 based PC, this time is in the order of 2.5 days. We have carried out 25,000 function evaluations, which is a typical value when measuring the performance of multiobjective algorithms using standard benchmarks, but this number is probably insufficient to obtain accurate fronts of solutions from a difficult problem such as DFCNT. However, increasing the number of evaluations can turn the runtime of the algorithms as impractical, what suggests that parallel techniques have to be considered if we intend to enhance the obtained results.

5 Conclusions and Future Works

We have analyzed seven multiobjective metaheuristics for solving DFCNT, the tuning of the DFCN broadcasting protocol for metropolitan MANETs. It is a complex real world problem, which uses Madhoc, a metropolitan MANET simulator, to model a realistic scenario that represents a shopping mall. This leads to a MOP having five decision variables and three objectives.

The chosen metaheuristics are representative of the state-of-the-art, and they include the well-known algorithms NSGA-II and SPEA2. Apart from these two methods, we have considered other metaheuristics: scatter search, cellular genetic algorithm, evolution strategy, differential evolution, and particle swarm optimization. The implementation of these algorithms and their application to solve DFCNT have been a join work of the members of the OPLINK project, involving research groups of four Spanish universities. The experiments carried out reveal that SPEA2 is the algorithm yielding the best results, according to the hypervolume metric. However, the metric values seem to indicate that all the algorithms produce fronts which are close together.

As lines of future work, more research has to be done to improve the Pareto fronts of the DFCNT problem. The study of fine-tuning the algorithms to enhance their performance and the use of parallel techniques to compute more function evaluations in a reasonable amount of time are matters of ongoing and future developments.

Acknowledgements. The authors are partially supported by the Spanish Ministry of Education and Science, and by European FEDER under contract TIN2005-08818-C04 (the OPLINK project, http://oplink.lcc.uma.es).

References

1. Conti, M., Giordano, S., Maselli, G., Turi, G.: MobileMAN: Mobile Metropolitan Ad Hoc Networks. In: Proc. of the Eight Int. IFIP-TC6 Conf, pp. 194–205 (2003)
2. Hogie, L., Guinand, F., Bouvry, P.: A Heuristic for Efficient Broadcasting in the Metropolitan Ad Hoc Network. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) KES 2004. LNCS (LNAI), vol. 3213, pp. 727–733. Springer, Heidelberg (2004)
3. Glover, F.W., Kochenberger, G.A.: Handbook of Metaheuristics. Kluwer, Dordrecht (2003)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans. on Evolutionary Computation 6(2), 182–197 (2002)
5. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical report, Swiss Federal Inst. of Technology (2001)
6. Coello, C.A., Lechuga, M.S.: MOPSO: A proposal for multiple objective particle swarm optimization. In: CEC 2002, pp. 1051–1056 (2002)
7. Nebro, A.J., Luna, F., Dorronsoro, B., Alba, E., Beham, A.: AbYSS: Adapting Scatter Search to Multiobjective Optimization. Technical Report ITI-2006-2, Dpto. Lenguajes y Ciencias de la Computación (2006)
8. Alba, E., Dorronsoro, B., Luna, F., Nebro, A., Bouvry, P.: A Cellular Multi-Objective Genetic Algorithm for Optimal Broadcasting Strategy in Metropolitan MANETs. Computer Communications 30(4), 685–697 (2007)
9. Bäck, T., Rüdolph, G., Schwefel, H.: A survey of evolution strategies. In: 4th Int. Conf. on Genetic Algorithms, pp. 2–9 (1991)
10. Storn, R., Price, K.: Differential evolution – a simple efficient adaptive scheme for global optimization over continuous spaces. Technical Report 95-012, Int. Compt. Sci. Inst., Berkeley, CA (1995)
11. Hogie, L., Guinand, F., Bouvry, P.: The Madhoc Metropolitan Adhoc Network Simulator. Université du Luxembourg and Université du Havre, France, Available at (2005), `http://www-lih.univ-lehavre.fr/~hogie/madhoc/`
12. Knowles, J., Corne, D.: The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation, CEC, pp. 9–105 (1999)
13. Luna, F., Nebro, A., Dorronsoro, B., Luna, F., Alba, E., Bouvry, P., Hogie, L.: Optimal Broadcasting in Metropolitan MANETs Using Multiobjective Scatter Search. In: Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H. (eds.) EvoWorkshops 2006. LNCS, vol. 3907, pp. 255–266. Springer, Heidelberg (2006)

14. Glover, F., Laguna, M., Martí, R.: Scatter Search. In: Advances in Evolutionary Computing: Theory and Applications, pp. 519–539. Springer, New York (2003)
15. Alba, E., Tomassini, M.: Parallelism and evolutionary algorithms. IEEE Trans. on Evolutionary Computation 6(5), 443–462 (2002)
16. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms – A Comparative Study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature - PPSN V. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)
17. Kukkonen, S., Deb, K.: A fast and effective method for pruning of non-dominated solutions in many-objective problems. In: Runarsson, T.P., Beyer, H.-G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) Parallel Problem Solving from Nature - PPSN IX. LNCS, vol. 4193, pp. 553–562. Springer, Heidelberg (2006)

The Influence of Data Implementation in the Performance of Evolutionary Algorithms

Enrique Alba[1], Edgardo Ferretti[2], and Juan M. Molina[1]

[1] Universidad de Málaga, Spain
{eat,jmmb}@lcc.uma.es
[2] L.I.D.I.C. Universidad Nacional de San Luis, Argentina
ferretti@unsl.edu.ar

Abstract. In this paper we study the differences in performance among different implementations, in Java, of the data structures used in an evolutionary algorithm (EA). Typical studies on EAs performance deal only with different abstract representations of the data and pay no attention to the fact that each data representation has to be implemented in a concrete programming language which, in general, offers several possibilities, with differences in time consumed, that may be worthy of consideration.

1 Introduction

Evolutionary algorithms (EAs) are stochastic search methods that have been successfully applied to a wide variety of search, optimization, machine learning, and design problems, and not only in research but also in industrial and commercial problem solving activities. These methods are inspired by the evolution of biological populations and the natural or environmental selection of the fittest individuals. Individuals are candidate solutions to the problem, populations are pools of candidate solutions, evolution is simulated by applying some operators (crossover, mutation,...) onto the population, and selection is performed stochastically with the aid of a fitness function. Consequently, an EA implementation always involves both an individual and a population representation.

The aim of this paper is to draw attention to the importance of the particular implementation, of the data structures used in an EA, once a problem encoding has been chosen. We refer to the data types, offered by a programming language, that we use to implement the abstract data types selected for representing the data used by the algorithm. This is the last step in the implementation process of an EA, and is not an insignificant issue considering the importance of execution time in real-world applications and in building algorithm components for software libraries. In our opinion, researchers (see [6]) have not paid sufficient attention to this subject. Also, we give some guidelines for measuring the performance and evaluating alternative implementations of data representations, in the line of [2], [3], and [5].

To achieve this, we have chosen as a running example a steady-state genetic algorithm solving the *One-Max* problem. Genetic algorithms (GAs) are the most

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 764–771, 2007.
© Springer-Verlag Berlin Heidelberg 2007

popular type of EA. They are typically implemented as a computer program in which individuals are represented as binary strings (chromosomes) of 0s and 1s —non binary encodings are also possible—, populations are represented as mathematical bags of chromosomes —bags admit different implementations—, and evolution is performed by recombination followed by mutation. The One-Max problem is a recurrent problem in EA literature since its simplicity makes it very suitable for focusing on the algorithm and not on the problem.

Our contribution is to shed some light on the software part of these algorithms, an often missed item in a vast literature worried about the applications and not about the implementations of the algorithms. In computer science, this is a particularly important issue, and research here can benefit the whole community of algorithm programming.

The paper is organized as follows. A description of the One-Max problem along with the steady-state GA is given in Sect. 2. Alternative implementations based on different Java representations for chromosomes and populations are presented in Sect. 3. An experimental study is reported in Sect. 4. The obtained results are analyzed with statistical tools in Sect. 5. Finally, in Sect. 6, we outline some conclusions and suggest future research directions.

2 The Problem Description and the Algorithm

The One-Max [1] or "counting ones" problem is a simple problem which consists of maximizing the number of 1s in a bit string. Formally, this problem can be described as the problem of finding a binary string $\bar{x} = (x_1, \ldots, x_N)$ which maximizes the function $F(\bar{x}) = \sum_{i=1}^{N} x_i$. Although it has a trivial solution $\bar{x} = (1, \ldots, 1)$, as said above, it is a recurrent problem in the EA literature, and therefore we have selected it for our study.

Here we focus on a steady-state GA described in Algorithm 1. This algorithm simulates evolution starting from an initial randomly generated population of chromosomes, $P(0)$, and proceeding, step by step, replacing the worst chromosome by a newly generated one, until a termination condition is reached. At each step t the algorithm produces a new chromosome ch by applying first a single point crossover (spx), with probability cp, to two parents, $p1$ and $p2$, selected from the population by binary tournament —based on their fitness—, and then mutation over the offspring ch', with probability mp. Chromosome ch replaces the worst chromosome of the population only if it is fitter. This way the population evolves until a solution is found or a fixed number of steps, max_steps, is reached.

In order to apply the above GA to the One-Max problem we need a representation for candidate solutions $\bar{x}$ (chromosomes) and a representation for populations of candidate solutions. We also need to fix the parameters: cp, mp, pop_size, and the maximum number of steps, max_steps, that will be used as a termination condition.

Algorithm 1. Simple steady-state genetic algorithm

```
SGA(cp, mp, pop_size, max_steps):
t ← 0;
P(t) ← generateRandomPopulation(pop_size);
evaluate(P(t));
while (not terminationCondition) do
   p1 ← selectBinaryTournament(P(t));
   p2 ← selectBinaryTournament(P(t));
   ch′ ← spx(p1, p2, cp);
   ch ← mutate(ch′, mp);
   evaluate(ch);
   P(t + 1) ← replaceTheWorst(ch, P(t));
   t ← t + 1;
end while
```

3 Alternative Data Structures

In order to program the described GA to solve the One-Max problem, it is necessary to choose a programming language for implementing the different procedures involved and to select the representations for individuals and populations within the data structures offered by the language. This choice has undoubted influence on the performance of the resulting program. Nevertheless, our aim in this paper is to highlight differences of performance due to alternative representations in a language, so we have chosen a popular language such as Java which offers many possibilities for data implementation.

We have focused on both small and large structures. A natural representation for candidate solutions to the One-Max problem is an array of 0s and 1s, but these values are susceptible to different representations in Java. We have selected a short (byte, 8-bits), an intermediate (int, 32-bits) and a long (double, 64-bits) data structure. Populations require a more complex (easy to traverse) data structure which lets us select and replace elements in an efficient way. Java offers several possibilities: `array`, the `Vector` class, and different implementations of a list. `array` is the most efficient structure, but the `Vector` class implements a growable array and can be used in programs like an array. To obtain programs as similar as possible we have used arrays and vectors in this study.

Combining all the above options the following six different implementations were obtained for the steady-state GA applied to the One-Max problem: *ga*01, *ga*02, and *ga*03 (first group), with `array` of individuals for populations and `array` of `byte`, `int`, and `double`, respectively, for individuals, and *ga*04, *ga*05, and *ga*06 (second group), with `Vector` for populations and `array` of `byte`, `int`, and `double`, respectively, for individuals. These are the programs whose performances are compared in the following sections. They differ only in the classes which implement the data structures storing the population and the individuals.

To study the differences among these programs we cannot use a theoretical analysis like the asymptotic count in the worst case, because all programs

implement the same algorithm and this analysis gives the same value $O(ga0i) = O(max_steps) \times max(O(pop_size), O(chrom_length))$, for $i = 1, \ldots, 6$. Nevertheless there should be differences among the programs because they differ in some methods that depend on the data implementation, particularly those methods for getting, setting or replacing an individual in a population, setting the fitness, and computing some statistics. These methods are called many times during the execution of a program, so the differences should become perceptible, at least between programs that use different implementation for the population.

4 Experimental Results

To confirm the above differences and to obtain some more subtle ones, we designed an experiment following [2]. We choose sixteen different problem instances corresponding to bit sequences (chromosomes) ranging from 64 to 1024 bits and fixed some algorithmic factors such as the population size (512 individuals), the crossover probability ($cp = 0.8$), and the mutation probability ($mp = 1.0/gn$ where gn is the number of genes in a chromosome), using standard values. After testing different values, the maximum number of generational steps was set to 1,000,000, which reported execution times near one minute (per execution) for the longest chromosomes. For each program $ga0i$, $i = 1, \ldots, 6$, and each problem size, we made 100 independent executions. Programs were run on the same machine[1] with the same compiler[2].

Table 1 shows the average run time of each program $ga0i$, for each problem size, in the 100 independent executions. In italics and boldface we can see the best times, and only in italics are the best times for programs which use a Java `Vector` based implementation for populations. As we can see, $ga02$ achieves the best results for 11 of the 16 instances, and $ga01$ for the remaining 5 instances. These results agree with the ones we expected to obtain for two main reasons:

1. There are two sets of arithmetic instructions in the Java Virtual Machine ([4]): one for integral types and another for floating-point types. They two use a 32-bit representation for operands in the operand stack, but `byte` operands have to promote to `int` values and the results have to be truncated to `byte` values again, while `double` values need two entries in the stack. This implies an extra overhead for reading and writing these values.
2. The Java `Vector` class, which represents a growable array of `Object`, uses a more complex implementation than the `array` of individuals and, consequently, the access methods are less efficient than when using the array ones. Also, retrieving an element involves some extra conversion operations from `Object` to the class that represents an individual.

For these reasons we expected $ga02$ to be the best in the group of programs that use an `array` representation for populations, $ga05$ to be the best in the other group, and each program in the first group to be better than the corresponding

[1] PC with Intel Pentium IV at 3.2 GHz, and 1 GB of RAM under Suse Linux 9.2.
[2] jdk 1.5.0.

Table 1. Average run time in seconds (100 executions)

Genes	ga01	ga02	ga03	ga04	ga05	ga06
64	0.3048	***0.3045***	0.3066	*1.0039*	1.0442	1.0408
128	0.9188	***0.8044***	0.9954	2.9342	2.8531	*2.6956*
192	2.5322	***2.1483***	2.2702	5.5811	*5.0032*	5.2692
256	4.3436	***4.1893***	4.2422	8.9356	*7.5753*	8.1553
320	6.5268	***6.2985***	6.7672	12.6860	*11.8308*	12.6346
384	***8.4674***	8.7704	8.9177	*15.4423*	17.6163	16.7914
448	***10.3140***	11.4702	10.7925	21.4521	22.5184	*20.2845*
512	13.2137	***12.4578***	13.1131	*24.3727*	24.5962	26.4751
576	***14.9605***	15.9595	15.4546	*29.4724*	29.9579	30.0714
640	***16.4292***	17.0087	17.0893	*35.3733*	36.5475	36.9828
704	***18.7165***	19.2592	19.0375	*37.6200*	39.6952	42.1398
768	20.7822	***20.1789***	21.1464	*44.6203*	45.7112	45.9348
832	22.7914	***22.7132***	23.5487	*48.7511*	51.4740	50.9820
896	24.6427	***24.3231***	25.0002	*54.5354*	56.4083	56.4753
960	26.4887	***26.3901***	27.0381	*59.0913*	60.8631	62.9465
1024	28.6279	***28.4375***	29.0843	*65.0828*	65.9349	67.3260

one (with the same chromosome implementation) in the second group. Nevertheless, these expectations were fulfilled only partially, because $ga01$ was faster than $ga02$ in 5 cases and $ga05$ only produced the best results of the second group in 3 cases, out of 16, against $ga04$ which is the best of this group in 11 cases.

5 Statistical Analysis

To obtain more general and better founded conclusions we analyzed the above results with statistical tools. These results show a linear relationship between the problem size (chromosome length) and the time consumed by each program. To confirm this fact we fitted a straight line,

$$time = a \times chrom_length + b,$$

to each program, by applying the linear least squares fitting technique. Adjustment parameters are presented in Table 2 together with the Pearson's coefficient of correlation r, and its square R^2, and Figs. 1 and 2 show the fitted straight lines for each program $ga0i$, in the first and in the second group respectively, with error bars of 5% at each point. These lines have been moved in order to avoid them overlapping. By analyzing these results we can conclude that all linear models fit well to their respective algorithms ($r > 0.99$ in all cases), and comparing the slopes it can be seen that programs which use vectors of individuals will consume more than twice the time consumed by the programs which use arrays of individuals. This confirms what was pointed out at the end of Sect. 3: `array` of individuals works better than `Vector`, and the experiments show that programs which use `Vector` consume more than 100% of the time needed by the programs which use `array`.

Besides, it is necessary to ensure that the results obtained in the experiment are not influenced by random noise. To do this we tried to apply an ANOVA to

Table 2. Linear models

Program	slope a	intercept b	R^2	r
$ga01$	0.0307	−2.9666	0.9977	0.9988
$ga02$	0.0307	−2.8925	0.9955	0.9977
$ga03$	0.0314	−3.0309	0.9975	0.9987
$ga04$	0.0682	−7.9323	0.9897	0.9948
$ga05$	0.0708	−8.5612	0.9903	0.9951
$ga06$	0.0721	−8.8594	0.9898	0.9949

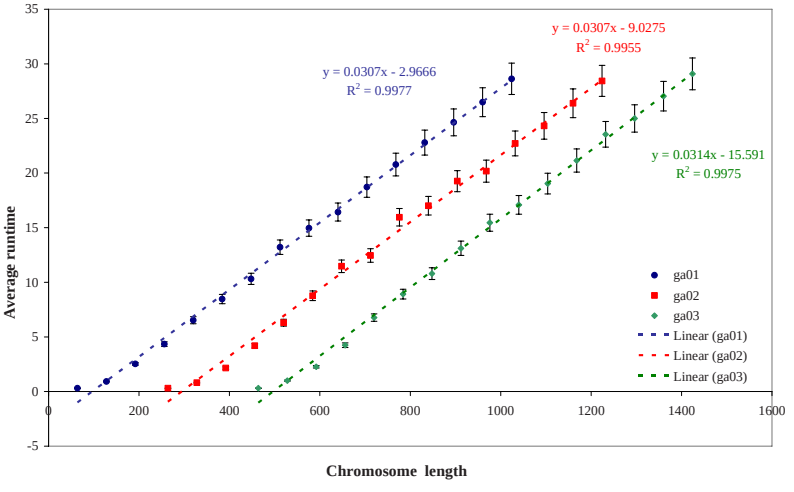


Fig. 1. GAs with `array` representation for populations

the samples obtained for each chromosome length, but the Kolmogorov-Smirnov test for normal distribution failed and it also failed the tests for homogeneity of variances. So applied the Kruskal-Wallis non-parametric analysis, and the Dunn post test. The obtained results were the following:

- For all tested chromosome lengths, differences between the first and the second group of programs were significant ($p < 0.001$).
- Differences between programs in the same group were not significant for 13 lengths (64, 128, 192, 256, 320, 384, 448, 512, 576, 768, 832, 896, and 960).
- For length 640, differences between $ga01$ and $ga03$ were significant ($p < 0.05$).
- For length 704, differences between $ga04$ and $ga06$ were significant ($p < 0.05$).
- For length 1024, significant differences were observed between $ga01$ and $ga03$, and $ga02$ and $ga03$ (both with $p < 0.001$); also between $ga04$ and $ga05$ ($p < 0.05$), $ga04$ and $ga06$ ($p < 0.001$), and between $ga05$ and $ga06$ ($p < 0.051$).

Finally, in Table 3 (left), we show the programs with the best median and the best average of both groups, and, in Table 3 (right), the programs of the second group which obtained the best median and the best average within this group.

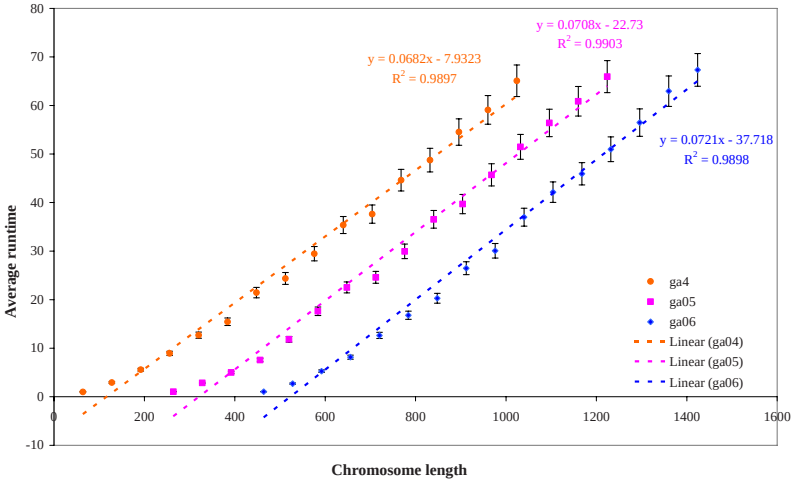


Fig. 2. GAs with `Vector` object representation for populations

Table 3. Best programs: global (left) and inside the second group (right)

Genes	average	median
64	*ga02*	*ga02*
128	*ga02*	*ga02*
192	*ga02*	*ga02*
256	*ga02*	*ga01*
320	*ga02*	*ga02*
384	*ga01*	*ga01*
448	*ga01*	*ga01*
512	*ga02*	*ga02*
576	*ga01*	*ga01*
640	*ga01*	*ga01*
704	*ga01*	*ga01*
768	*ga02*	*ga02*
832	*ga02*	*ga02*
896	*ga02*	*ga02*
960	*ga02*	*ga02*
1024	*ga02*	*ga01*

Genes	average	median
64	*ga04*	*ga04*
128	*ga06*	*ga05*
192	*ga05*	*ga04*
256	*ga05*	*ga05*
320	*ga05*	*ga04*
384	*ga04*	*ga04*
448	*ga06*	*ga06*
512	*ga04*	*ga04*
576	*ga04*	*ga05*
640	*ga04*	*ga05*
704	*ga04*	*ga04*
768	*ga04*	*ga05*
832	*ga04*	*ga04*
896	*ga04*	*ga06*
960	*ga04*	*ga04*
1024	*ga04*	*ga04*

These results confirm what was claimed in Sect. 4: *ga*02 and *ga*04 are the best of all implementations and the best of the second group, respectively.

6 Conclusions and Future Work

In this article we have studied the influence in performance of alternative Java data structures for a basic EA. We have designed an experiment for comparing six different implementations of a steady-state GA, grouped into two groups depending on the population representation (`array` or `Vector`) and distinguishing

within each group by the individual representation. The results obtained were the following: programs in the second group consumed 100% more time than their equivalent in the first group and, according to a statistical analysis of these results, differences among programs in the same group are not significant in most of the chromosome lengths. Only for the longest chromosome (1024 bits) can significant differences be observed in favor of *ga*01 and *ga*02 in the first group, and *ga*04 in the second group. The conclusions of this study for steady-state GAs are the following:

- The data structure used for population representation significantly affects the time consumed by the (Java) implementation.
- The type used for genes does not have a noticeable influence, at least for the considered chromosome lengths. Maybe, for longer chromosomes this influence will become perceptible.

As we said in the introduction, more attention needs to be paid to data implementation in works about EAs. This paper is just a first step in this direction and we will try to continue working with other algorithms, other problems, and other programming languages. Our aim is to obtain some guidelines about which structures fit which algorithms best, and to what degree it depends on the problem.

Acknowledgements

This work has been partially funded by the Spanish Ministry of Education and the European FEDER under contract TIN2005-08818-C04-01 (the OPLINK project, http://oplink.lcc.uma.es).

References

1. Ackley, D.H.: A connectionist machine for genetic hillclimbing. Kluwer, Dordrecht (1987)
2. Alba, E., Luque, G.: Measuring the Performance of Parallel Metaheuristics. In: Parallel Metaheuristics, ch.2, pp. 43–61. Wiley, Chichester (2005)
3. Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.G.C., Stewart, W.R.: Designing and Reporting on Computational Experiment with Heuristics Methods. Journal of Heuristics 1, 9–32 (1995)
4. Lindholm, T., Yellin, F.: The $Java^{TM}$ Virtual Machine Specification. Addison-Wesley, London, UK (1996)
5. Rardin, R.L., Uzsoy, R.: Experimental Evaluation of Heuristic Optimization Algorithms: A Tutorial. Journal of Heuristics 7, 261–304 (2001)
6. Rothlauf, F.: Representations for Genetic and Evolutionary Algorithms. Springer, Heidelberg (2006)

Heuristic Approach to Conflict Problem Solving in an Intelligent Multiagent System

Witold Jacak and Karin Pröll

Dept. of Software Engineering at Hagenberg
Upper Austrian University of Applied Sciences
Softwarepark 11, A 4232 Hagenberg, Austria
jacak@fh-hagenberg.at, proell@fh-hagenberg.at

Abstract. The paper presents a method that allows an intelligent multiagent system to coordinate and negotiate their actions in order to achieve a common goal. Each individual agent consists of several autonomous components that allow the agent to perceive and react to its environment, to plan and execute an action, and to negotiate with other agents in an intelligent manner. A heuristic approach for conflict solution is presented, which is used for coordination of a society of independently acting agents in common environment. The application of the method is shown on an intelligent multiagent robotic system realizing complex transfer operations simultanously.

Keywords: multi agent systems, intelligent robotic agent, heuristic conflict problem solution.

1 Introduction

Intelligent agents are a new paradigm for developing complex system applications. The most powerful tools for handling complexity in system development are modularity and abstraction. Industrial application of agent technology were among the first to be developed, in such domain as process control, manufacturing and robotic systems. The agent systems base on autonomous software-hardware components (technical agents) that cooperate within an environment to perform some task. This paper presents a heuristic approach for conflict management in an intelligent multiagent system, which is used to coordination of a society of independently acting agents in common environment [1,2,6,7,9,10,12,13].

1.1 Multiagent System

A multiagent system is a group of cooperating agents A_i acting in common environment. Cooperation between agents has a different focus. The main task is to solve complex problems by using cooperation and communication mechanism with other agents or external resources. For the communication among agents the communications platform have to be established and communication protocols

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 772–779, 2007.
© Springer-Verlag Berlin Heidelberg 2007

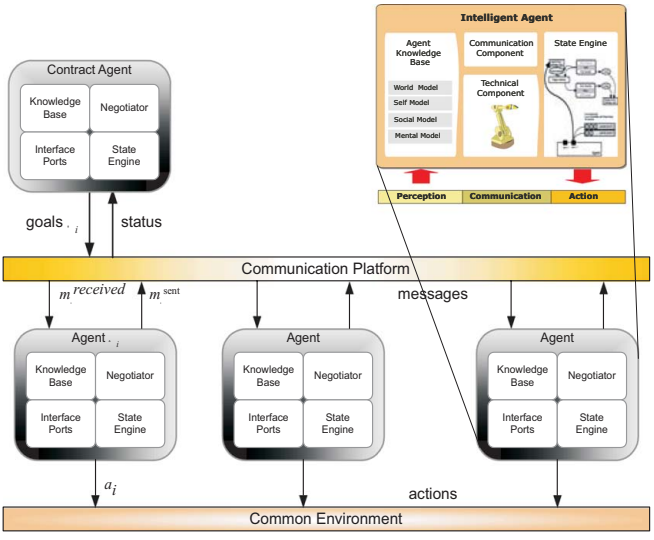


Fig. 1. Multiagent system

have to be defined in order to ensure that agents can exchange information. The behavior of the overall multiagent system can be observed as the set of actions performed by each individual agent. The special contract agent determines the goal for each agent. Each agent A_i follows its own specific goal g_i. The set of all goals G is quasi ordered by a priority relation (reflexive and transitive), and can change in the time. Each agent has its own goal g_i and an autonomous behavior, which is a result of its observations, its knowledge and its interactions with other agents [1,2,3,4]. A multiagent system is presented in Figure 1.

1.2 An Intelligent Agent

Each intelligent agent consists of several cooperating units such as: KB - knowledge base, Neg - negotiator, Int - relay ports (the interface) the set of agent input and output and E - state engine that describes the behavior of an agent. The i-th technical agent can be represent by a tuple

$$A_i = (KB_i, Int_i, Neg_i, E_i) \tag{1}$$

The behavior of an agent is specified by state engine E. State engine can be represent by a tuple

$$E_i = (Ac_i, S_i, M_i, \tau_i, \delta_i, \lambda_i, \kappa_i) \tag{2}$$

where:

- Ac_i is the action set of agent. The action represents the possible activities of the agent,
- M_i is the set of incoming and outgoing messages,

- S_i is the set of states of the agent. The state can be passive or active. The active state has finite advance time defined by π_i (time advance function),
- the trigger function $\delta_i : S_i \times Ac_i \rightarrow S_i$ determines the state transition,
- the response (output) function $\lambda_i : S_i \rightarrow M^{sent}$ generates the possible output messages connected with the state,
- the conflict function $\kappa_i : M_i^{received} \times S_i \rightarrow \{T, F\}$ determines the feasibility (conflict freeness) of state based on incoming messages. When $\kappa_i(m_i^{received}, s_i) = T$ it means that state s_i is in direct conflict with another agent state [8,9,13].

2 Goal Reachability Problem

The goal g_i of an agent determinates the set of final states $S_i^{final} \subset S_i$ of A_i in current moment of the time. The main goal of a multiagent system is to realize all of the goals from set G, which are distributed to the individual agents. The problem of goals realization for all active agents appears to be a problem of *reachability* of the set of final states from the start state for each agent. In order to solve such reachability problem we are going to apply the graph searching procedure to the state transition graph of state set S_i. The way of expanding the graph will depend on the form of the cost function using to evaluate each node. As the evaluation function we can use the sum of the cost function $c(s_i^{start}, s_i^c)$ and a cost estimate heuristic function $h(s_i^c, s_i^{final})$, the estimated cost between agent current state and terminal state. Using the standard A^* procedure we can find the state path

$$s_i^* = (s_i^{start}, s_i^c(1), .., s_i(k), .., s_i^{final}) \tag{3}$$

from current state to goal state which includes only feasible states.
The problem of optimal state path s_i^* planning for i-th agent amounts to finding a sequence of actions

$$ac_i^* = (a_i(1), .., a_i(K)) \tag{4}$$

such that: the terminal state is achieved, every state $s_i(k)$ for $k = 1, .., K$ is conflict free and the sequence ac_i^* minimizes the cost function $e_i = c_i + h_i$.

Fact 1. *If for all agent states, no conflict occurs then the global trajectory s^* is composed from the s_i^* state trajectories of each active agent.*

As the individual behavior of all agents involved in own goal achievement cannot be predicted in advance, the states of two or more agents can be in direct conflict with one another and the achievement of the goal is endangered. In order to resolve a conflict situation, a negotiation process among all conflict parties has to take place. The result of the negotiation should be a solution, which results in goal achievement for all involved agents [2,3].

2.1 Sequential Synchronized Reachability Solution

The conflict between the goal states from the set S_i^{final} and states of another agent causes that for example a solution for the reachability problem does not

exists. To avoid this problem we propose the sequential decomposition of the global search problem into coordinated sequential search for each agent separately. It means that the not existing simultaneous solution will be substituted by a sequence of partial solutions distributed in time.

Fact 2. *If for each agent there exists an optimal trajectory, which achieves the goal state separately conflictless in start state of each agents, and an additional trajectory, which states are not in conflict with each state from optimal trajectory of other agents then the global reachability problem can be decomposed into n sequential synchronized problems such that each agent realizes part of it's optimal trajectory, when all other agents move to the conflicts free states sequentially [13].*

The global goal will be reached by sequence of realization of local goals for each agent (Figure 2).

3 Heuristic Conflict Problem Solving

The state machine model of the agent behavior can be applied to solve the conflict avoidance problem by substitution of the concurrent performed planning and realization of actions of agents group by the sequentially performed one or more steps planning and realization of the action of several agent from agent group. Such round robin procedure for time synchronization of agent action leads to concurrent realization of actions with k-step delay and to solve the conflict avoidance by using the sequence time synchronized partial solution of goal achievement. Let $AGENT(t)$ be a list of agents ordered by values of priorities of their goals in time moment t. We will look for the best action of each agent in moment t, which does not cause the conflict and leads to final states set. For this purpose we shall exploit the state transition graph of the each agent generated implicitly by applying the transition function as production rules sequentially for each agent from list $AGENT(t)$.

3.1 Sequential Coordination and Planning Algorithm

The first agent from the list generates the new part of state's path in following steps:

Step 1. Agent requests the message about current states of other agents in common environment, and establishes the own conflict states.

Step 2. Based on the conflict states and its own status, agent recognizes its own current situation and establishes the parameters for graph search algorithm, such as type of evaluation function and type of state feasibility testing function.

Step 3. Action planner starts the state-graph searching algorithm from its current state with previously established evaluation and conflict freeness testing functions. The searching stops if the goal state is reached or if the OPEN set is empty or if the OPEN set has more as N elements.

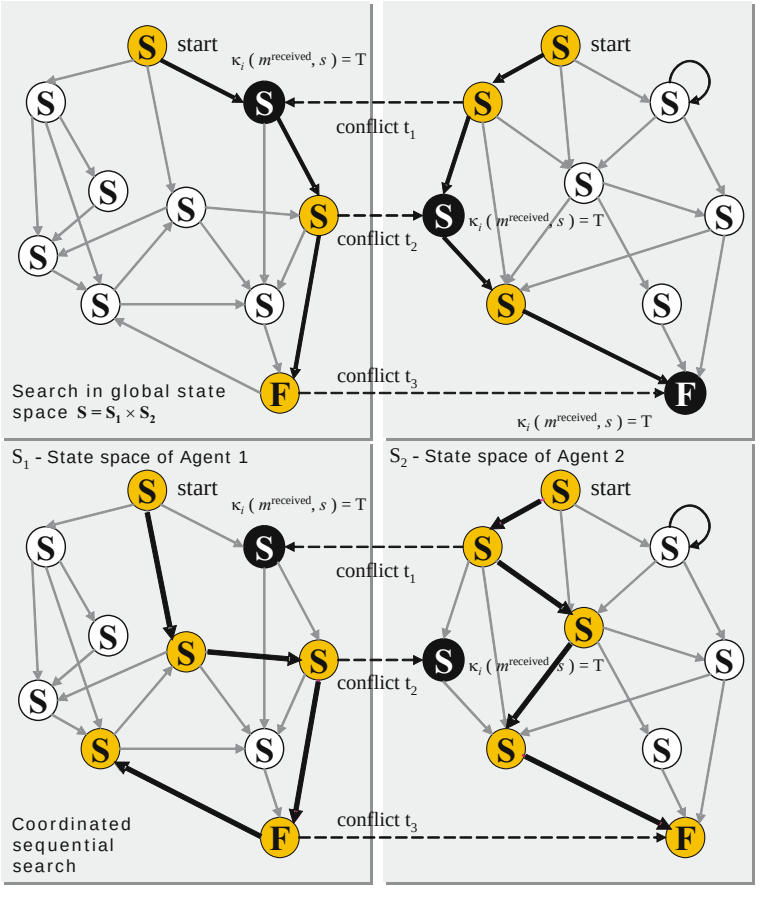


Fig. 2. Optimal trajectory, but conflict situations above and coordinated sequential search for conflict avoidance below

Step 4. The temporary path of states is calculated i.e. $path^{tem} = s_i^c \rightarrow s_i^b$ where s_i^b is the best state in CLOSE set. Depending of the length and conflict freeness of this path the new state chosen and actions sequence is realized.

Step 5. The new current state is analyzed and the adequate message is send to other agents. The message includes the information if agent has achieved its goal and/or if the conflict occur. Negotiator *Neg* trys to perform change the priority level of agent goal, based on current distance to final state. This leads to change the order of AGENT list for the next step. The new list is sent to each agent.

With sending the message agent ends his activity and the next agent starts the path planning. In each interval of time only one agent is working. The other agents are waiting for message Action **done** to come [11,12,13]. The overall negotiation process is shown in Figure 3.

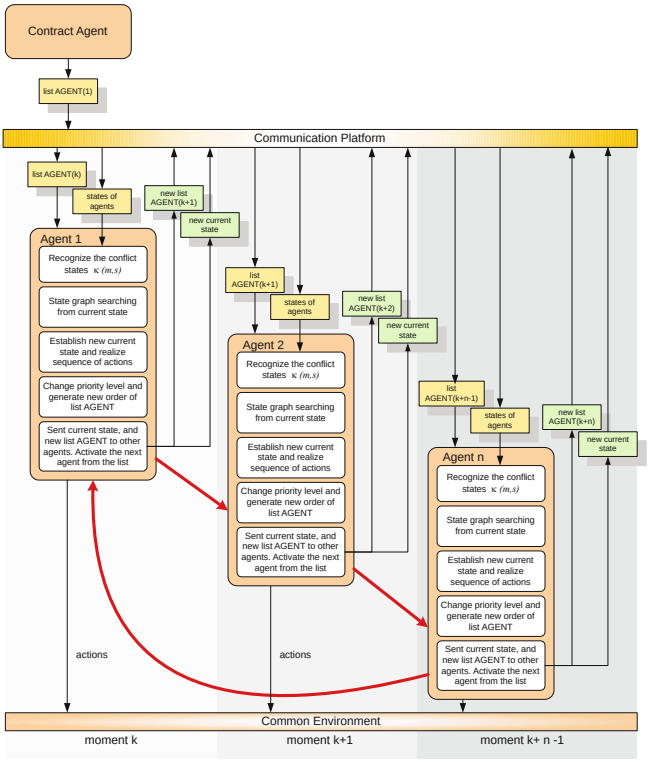


Fig. 3. Action planning and conflict solving cycle

4 Application in Intelligent Robotic Agents System

This section presents an intelligent multiagent robotic system as an application of the general multiagent system approach, which is used to control and coordinate a community of independently acting robotic agents in partially known environment.

Each robotic agents contains two main units: the software unit (intelligent controller) and the hardware unit (robot). The general goal of a multiagent system is to solve all tansfer tasks, obtained from other parts of the manufacturing system. Our considerations will focus on groups of service robots realizing their goals with different motions. Goal distribution and supervising of goal execution is done by the management level responsible for cooperation and negotiation between robotic agents on the operational level.

4.1 Intelligent Robotic Agent

Robot kinematics with n DOF can be described as FSM Model in Joint Space Q. State of the robotic agent is $q = (q_1, .., q_n)$ and the action can be modelled as

$\delta q = (\delta q_1, .., \delta q_n)$. If the distance between two agents is less than a safety zone it will be denoted as conflict between these agents. The feasibility of the agent current state $q(i)$ can be expressed as the geometrical distance between states $q(j)$ of different agents $\rho(q(i), q(j))$. If $\rho(q(i), q(j)) < \epsilon$ then $\kappa_i(m^{receivedfromj}, q(i)) = T$ [1,2,5].

4.2 Example

Figure 4 presents a full conflict situation between Agent 1 and Agent 2 in various and final positions. Concurrent planning and realization of actions of agents group without negotiation leads to full conflict in final positions. The agents do not achieve their own final states.

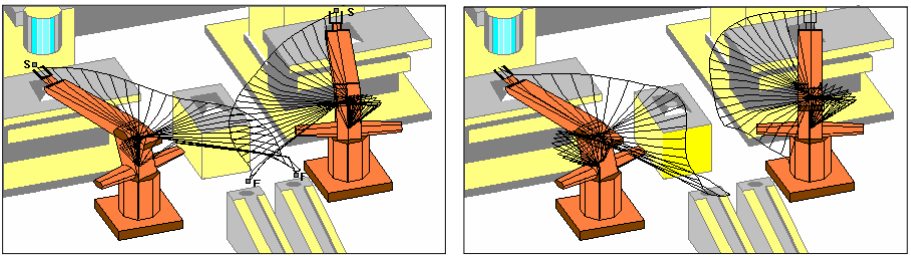


Fig. 4. In the left picture a full conflict between Agent 1 and Agent 2 in various positions, including the final state is shown. In the picture on the right concurrent planning and realization leads to not achieving final states.

Solving the conflict avoidance problem by substitution of the concurrent planning and realization of actions of robots group with the sequentially performed one step planning and realization of the action of several robot. The round robin procedure for time synchronization of agent action leads to concurrent realization of actions with k-step delay and to solve the conflict problem by using the sequence time synchronized partial solution of goal achievement. (Figure 5)

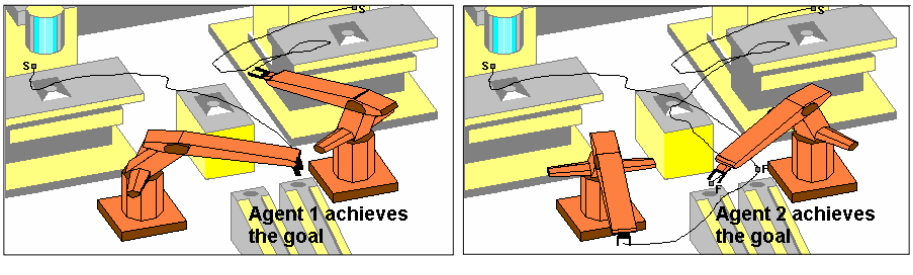


Fig. 5. Conflict situation resolved

References

1. Jacak, W., Pröll, K.: Multiagent Approach to intelligent control of robot, Robotics and Applications RA99, IASTED International Conference. Santa Barbara, USA (1999)
2. Jacak, W., Pröll, K.: Multiagent based intelligent control and internal communication in an autonomous system, SCI'99 and ISAS'99, World multiconference on systemics, cybernetics and informatics, Proceedings Volume III (1999)
3. Jacak, W., Pröll, K.: Multiagent Approach to Intelligent Control of Robot. In: Kopacek, P., Moreno-Diaz, R., Pichler, F. (eds.) EUROCAST 1999. LNCS, vol. 1798, Springer, Heidelberg (2000)
4. Brenner, W., Zarnekow, R., Wittig, H.: Intelligent Software Agents. Springer, Heidelberg (1998)
5. Jacak, W.: Intelligent Robotic Systems. Kluwer Academic/Plenum Publishers (1999)
6. Sandholm, T.: Automated Negotiation. Communications of the ACM 42(3), 84–85 (1999)
7. Sandholm, T.: Distributed Rational Decision Making. In: Weiß, G. (ed.) the textbook Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence, pp. 201–258. MIT Press, Cambridge (1999)
8. Andersson, M., Sandholm, T.: Sequencing of Contract Types for Anytime Task Reallocation. In: Sierra, C. (ed.) AMET 1998 and AMEC 1998. LNCS (LNAI), vol. 1571, pp. 54–69. Springer, Heidelberg (1999)
9. Weiss, G. (ed.): Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. The MIT Press, Cambridge, Massachusetts (1999)
10. Arai, S., Sycara, K.: Multi-agent reinforcement learning for planning and conflict resolution in a dynamic domain. In: Proceedings of the Fourth International Conference on Autonomous Agents, ACM Press, Multi-agent reinforcement learning for planning and conflict resolution in a dynamic domain (2000)
11. Brazier, F., Keplicz, B.D., Jennings, N.R., Treur, J.: Formal specification of multiagent systems: a real-world case. In: First International Conference on Multi-Agent Systems (ICMAS'95), AAAI Press, San Francisco, CA (1995)
12. Haigh, K.Z., Veloso, M.M.: High-level planning and low-level execution: Towards a complete robotic agent. In: Proceedings of the First International Conference on Autonomous Agents, ACM Press, New York (1997)
13. Pröll, K.: Intelligent Multi-Agent Robotic Systems: Contract and Conflict Management, PhD Thesis, Johannes Kepler University Linz /Austria (2002)

Optimal Placement of Sensors for Trilateration: Regular Lattices vs Meta-heuristic Solutions

J.O. Roa[1], A.R. Jiménez[1], F. Seco[1], J.C. Prieto[1], and J. Ealo[1,2]

[1] Instituto de Automática Industrial - CSIC
Ctra. Campo Real Km 0.2, 28500 La Poveda, Madrid, Spain
{javieroa,arjimenez}@iai.csic.es
http://www.iai.csic.es/lopsi
[2] School of Mechanical Engineering, Universidad del Valle,
Cali - Valle del Cauca, Colombia

Abstract. Location-aware applications, such as indoor robot navigation or human activity monitoring, require the location estimation of moving elements, by using ultrasonic, infrared or radio signals received from sensors deployed in the workplace. These sensors are commonly arranged in regular lattices on the ceiling. However, this configuration is not optimal for location estimation using trilateration techniques, in terms of positioning precision, maximum coverage and minimum singular cases. This paper shows how non-regular optimal sensor deployments, generated with a new meta-heuristic optimization methodology (Diversified Local Search - DLS), outperforms regular lattices for trilateration.

Keywords: Heuristic optimization, Diversified Local Search, Trilateration singularities, Optimal sensor locations.

1 Introduction

Location-aware systems find important applications in a diverse range of fields, such as outdoors vehicle guidance, reliable robot navigation or human activity monitoring. The most successful outdoor location solution is the well known Global Positioning System (GPS). However, there is not an equivalent solution for indoors. Some solutions developed for indoor use rely on ultrasonic, infrared or radio technology [1, 2], but a definitive system has not been accepted yet, due to several technical reasons and cost constrains. Indoor Local Positioning Systems (LPS) is an important and open research field.

Most location systems use trilateration techniques for position estimation [3–5]. They estimate the unknown coordinates of a mobile, given the known coordinates of a group of reference points (sensors, antennas or satellites), and the distances between each sensor and the mobile. The number of sensors and available distances has to be sufficient in order to find the unknowns by solving a nonlinear system of equations.

For proper trilateration, the sensor distribution in relation to the mobile location has to be favorable. Unfavorable deployments can induce singularities and

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 780–787, 2007.
© Springer-Verlag Berlin Heidelberg 2007

low precision in the position estimations. Therefore, it is important to know the right placement of sensors, in order to avoid bad configurations. The well known metric termed Dilution Of Precision (DOP), can be used to detect such configurations. The DOP is a dimensionless number, that decreases when the trilateration problem becomes better conditioned (good geometry among sensors). It can be expressed as the ratio of standard deviations between location estimations and measured distances [6, 7], given as:

$$DOP = \sqrt{\sigma_{x_k}^2 + \sigma_{y_k}^2 + \sigma_{z_k}^2}/\sigma_{d_k}. \tag{1}$$

There are two main trilateration techniques called spherical (ST) and hyperbolic (HT). ST uses absolute distances between sensors and mobile device, whereas HT uses differences of distances between sensors and mobile with respect to one reference sensor. Due to this fact, ST requires one sensor less than HT. In ST systems the singularities are generated when the available sensors are positioned in a straight line, whilst in HT systems it also happens when the mobile location is such that differences of distances becomes null. Therefore, according to the selected technique, it is possible to find different DOP values, singular areas (SA) and non-coverage areas (NC) which are regions where the available amount of sensors is not enough to solve the equation system. A representative case of singularity using ST is shown in figure 1. Figure 1a illustrates how a mobile moves from a non-singular place (A) to a singular place (B) where three sensors are aligned. Figure 1b shows an overhead view of figure 1a with an indication of DOP values, SA and NC areas. Figure 2 shows a case of singularity using HT. Notice that the same sensor distribution is used. However, DOP, NC and SA in HT are worse in comparison with the ST case (see figure 2b).

When the work area has obstacles, walls or an irregular shape, finding an optimal sensor deployment is a non-trivial task. It is possible to consider a regular distribution using a large amount of sensors in order to avoid SA [8]. However, this solution is not recommendable since it increases the cost of the system. Several 2D and 3D location system based on a trilateration technique have been developed. Some of these systems used regular sensor deployments [5, 9], but do not consider singularities or the low precision caused by aligned or coplanar sensors. As a result, these systems require a redundant amount of sensors. Another works have studied optimal sensor deployment for 2D and 3D scenarios [10, 11]. The optimal 2D solution consisted of three sensors forming an equilateral triangle and a fourth one at the center of the triangle; both sensors and mobile were on the same plane. 3D optimal solutions were based on sensors distributed on a unit spherical surface, the sensors were located on the vertices of 3D shapes called Platonic solids (tetrahedron, octahedron, etc). However, all these sensors distributions are not valid for a realistic deployment in indoor environments since do not have into account blockage of signals, free space installation, etc.

This work aims at demonstrating how optimal sensor deployments, generated with a meta-heuristic optimization strategy called Diversified Local Search (DLS), outperform regular arrangements, such as square or triangular lattices. DLS searches for optimal solutions providing maximum positioning accuracy

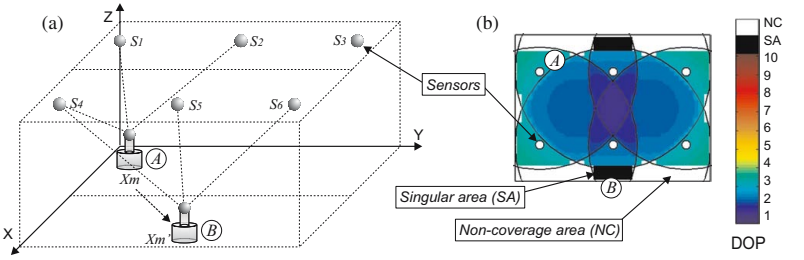


Fig. 1. A representative case of singularities in a ST system using a regular square lattice sensor deployment: (a) The mobile device moves from a place (A) without singularities to a singular place (B). (b) Top view with DOP values, NC and SA areas.

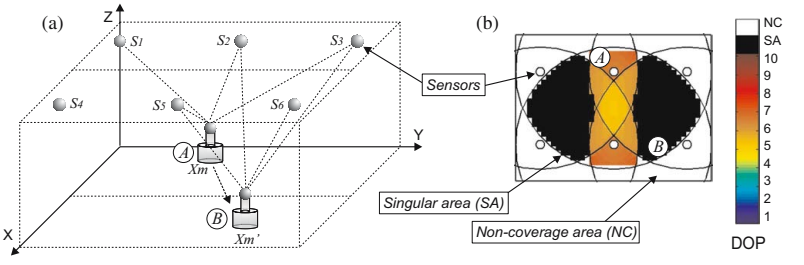


Fig. 2. A representative case of singularities in a HT system using a regular square lattice sensor deployment: (a) The mobile device moves from a place (A) without singularities to a singular place (B). (b) Top view with DOP values, NC and SA areas.

(low DOP) and maximum coverage with a few number of sensors (low cost). Realistic scenarios are considered; therefore sensors were on the ceiling in order to achieve Line-Of-Sight (LOS) of signals. Next section describes the methodology used to calculate and generate the best sensor deployments. In section 3, results are presented, showing examples of optimal deployment solutions applying ST and HT. Finally, a discussion and conclusions are given.

2 Methodology

2.1 Quality of Sensor Deployments

Two factors to describe the performance of sensor deployments were considered: **Precision** and **Non-Availability**. These factors were combined according to the following fitness function:

$$f(\Omega) = \underbrace{\overline{\mathrm{DOP}} \times \mathrm{K1}}_{Precision} + \underbrace{\mathrm{NAR} \times \mathrm{K2}}_{Non-Availability} \quad . \tag{2}$$

Where:

- Ω: It is given by the set $\{[x_i, y_i, z_i] : i = 1, ..., n\}$ of coordinates of all sensors included in the solution.
  $\overline{\mathbf{DOP}}$: It is the mean of DOP values over the desired localization area.
- **NAR**: It is the Non-Availability Ratio. It is given as: (NC+SA)/*Area*. Where, *Area* is the location area that we aim to cover with the positioning system.
- **K1 and K2**: These are weights to balance the level of importance desired for each term in the fitness function.

The fitness function (2) aims at maximizing precision and the available area for location of mobile device. Therefore, DOP and NAR must be reduced in order to find optimal sensor deployments. To perform the DOP, NC, SA and Area computations, the intersections in a square grid with 10 cm of resolution, over the desired localization area, were used as the evaluation points.

2.2 Methods to Find the Best Sensor Deployments

In equation (2), values for K1 and K2 were selected to get deployments solutions according to desired trade-off between precision and availability. These were K1=15 and K2=400 for both ST and HT cases. Using these weights, the best regular sensors deployment based on square and triangular lattices were found by changing uniformly inter-sensor distances (approx. from 0.5 to 2.5m). These solutions corresponded to regular deployments that produced the minimum value in (2). For meta-heuristic optimization, we start with an grouped initial distribution (very close sensors), the equation (2) was minimized as a multi-objective problem. To minimize it, the DLS optimization method [13] was used. It is based on a combination of Local and Tabu search. The aim of this search is to find the best DLS sensor deployments. Following this, a 2D cost function space between DOP and NAR was used to compare the performance of regular and DLS solutions. The cost of the positioning system is important and it is proportional to the number of sensors used. Therefore, a low number of sensors was kept constant during the search of best deployments. It allowed to observe the performance of regular and DLS deployments using few sensors.

Using the DLS method, different values for K1 and K2 yield to different optimal solutions in terms of DOP and NAR. In order to find the best possible solution for different K1 and K2 values, the Pareto Optimality Criterion (POC) was applied, where different factors are balanced in such a way that no improvement can be achieved by decreasing one factor without increasing another. Applying POC, a curve of solutions called *Pareto Frontier* was found, by minimizing the equation (2) using DLS method, changing the values of K1 and K2, in each search, from K1$\gg$K2 (the best DOP and the worst NAR) and vice versa.

3 Results

Experiments were defined for a square area (of 25 m^2) in order to consider a room with constant LOS propagation between the sensors and the mobile device, and

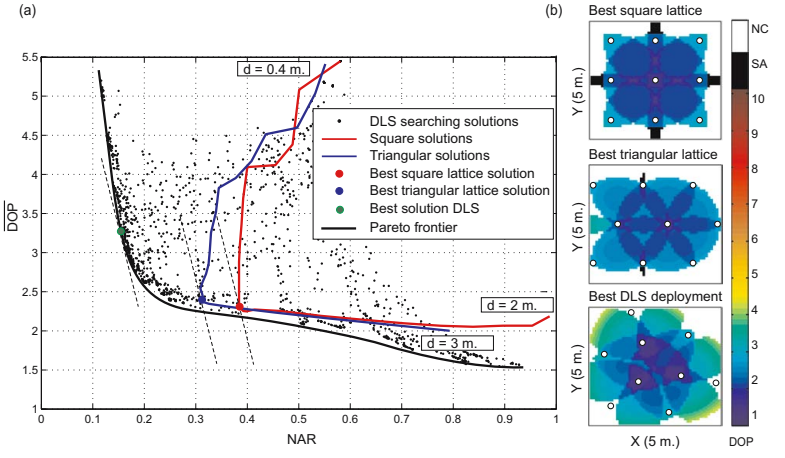


Fig. 3. Optimization results for a square area applying ST: (a) Cost function space, (b) Best sensor deployments

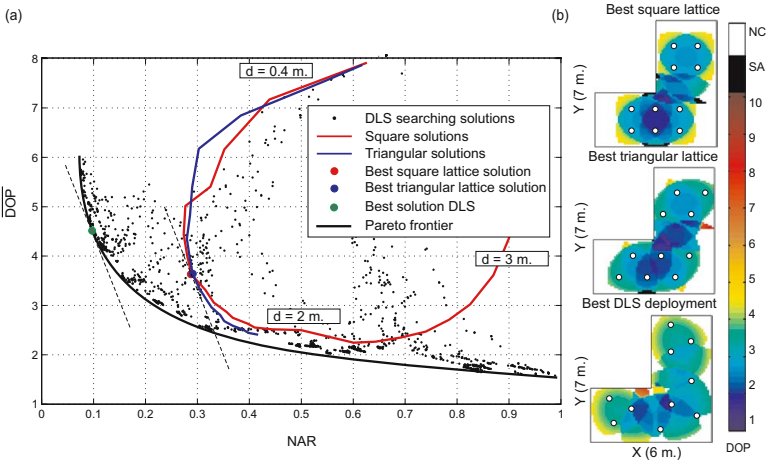


Fig. 4. Optimization results for a L-shaped area applying ST: (a) Cost function space, (b) Best sensor deployments

a L-shaped area (of 30 m^2) whose outer-corner walls causes blockage of signals. A realistic scenario with 2 m height was simulated. The sensor locations were on ceiling and the location area was on floor. The number of sensors was 9 and 14 for the square and L-shaped areas, respectively, in order to observe how deployments with few sensors can be improved. Considering a sensibility limited by the reception pattern of sensors at 45°, the link between the sensor and the mobile was available in a circumference of radius 2 m over the location area just below each sensor. First, it is shown a comparison among regular lattice and DLS solutions, using ST and then using the HT.

Table 1. Values of f(Ω), DOP and NAR for all the best sensor deployments found

	Spherical Tecnique (ST)						Hyperbolic Technique (HT)					
	Square room			L-shaped room			Square room			L-shaped room		
	f(Ω)	DOP	NAR	f(Ω)	DOP	NAR	f(Ω)	DOP	NAR	f(Ω)	DOP	NAR
Best square lattice	188.7	2.298	0.385	170.8	3.619	0.291	183.9	6.714	0.208	301.0	7.145	0.484
Best triangular lattice	159.2	2.405	0.308	169.8	3.626	0.288	195.8	6.570	0.243	309.2	7.248	0.501
Best DLS deployments	**110.4**	3.183	0.156	**108.8**	4.486	0.104	**157.4**	7.127	0.126	**254.8**	7.328	0.362

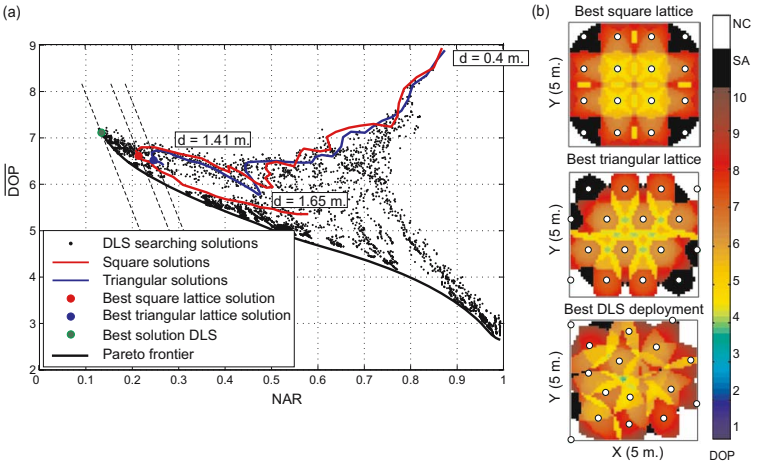


Fig. 5. Optimization results for a square area applying HT: (a) Cost function space, (b) Best sensor deployments

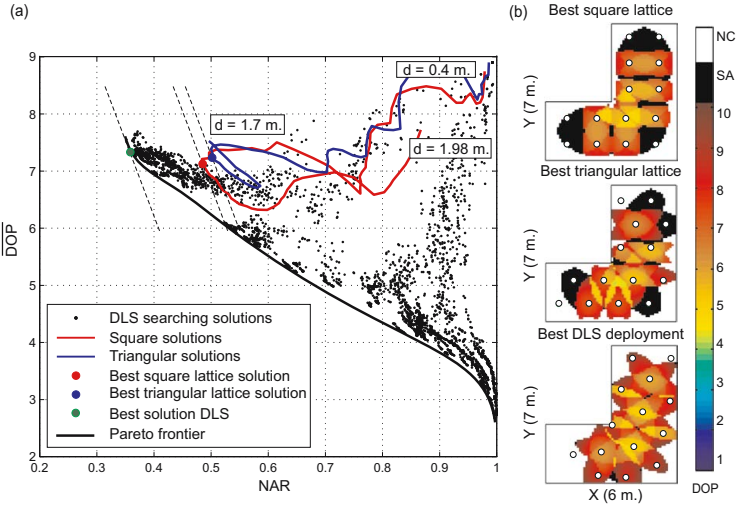


Fig. 6. Optimization results for a L-shaped area applying HT: (a) Cost function space, (b) Best sensor deployments

The performance curves between DOP and NAR for regular lattices and DLS solutions, applying ST, in the square and L-shape area, are presented in the figure 3a and 4a, respectively. The black trace is the *Pareto Frontier*, which was found with DLS method. Red and blue traces represent all of the solutions that were found for square and triangular sensor deployments, changing uniformly inter-sensor distances (d). The green, red and blue points are the best DLS, square and triangular found solutions, respectively. The dashed straight line represents the linear combination of precision and availability using the selected values for K1 and K2 (15 and 400, respectively) in the equation (2). The DLS solution was the best found. However, the best lattices solutions were suboptimal, a good DOP was achieved, but the NAR was poor for square and triangular lattices. The best selected sensor deployments are shown in figure 3b and 4b for square and L-shape area, respectively. White circles, black and white zones represent the sensors, SA and NC, respectively. The colored bar allows to identify the best (blue) and worst (red) DOP areas.

Applying HT method in both areas, the best found sensor deployments based on regular lattices present more NAR than the best DLS solution, keeping a similar DOP values. Therefore, DLS solutions remain as the preferable option. Figure 5 and 6 show the results for the square and L-shaped area, respectively. Figure 5b and 6b show that the best DLS deployments generated less SA than the best regular deployments, where generated SA were considerable. Besides in DLS deployment solution a sensor was exactly located in the out-corner of L-shape area, maximizing the availability. Values of $f(\Omega)$, DOP and NAR, for the best found sensor deployment, are summarized in the table 1.

4 Discussion and Conclusions

Regular sensor distributions generate singular areas and their coverage is poor. These problems appear when the positioning system uses few sensors and become bigger when it apply HT than ST. Optimization methods such as DLS allow that sensor distribution adapts to the shape of the location area. It was notable in the L-shaped area since using DLS the sensor locations become adaptable to the shape of area, avoiding obstacles and achieving more coverage. This adapted distributions are not intuitive. Therefore, these tools are necessary in order to get the best coordinates of sensor locations. Making a comparison between the found DLS *Pareto Frontier* and the curves found with regular deployments, it is observed that the best solutions were always obtained using DLS method, independently of the arbitrary selected values of K1 and K2. If the positioning system has a redundant amount of sensors, a deployment solution based on regular lattices is sufficient since good precisions and availability are achieved. Therefore, DLS solutions are useful only when it is desired to use few sensors in the system.

This work studied how an optimization process as DLS is able to find optimal sensor deployments that outperform regular distributions. This search strategy

avoids singularities and achieves better coverage with good positioning precision. These results are important for sensor network deployment with minimum infrastructure costs.

References

1. Priyantha, N.B., Chakraborty, A., Balakrishnan, H.: The Cricket Location-Support System. In: Proceedings of the 6th ACM International Conference on Mobile Computing and Networking, Boston MA USA, pp. 32–43 (2000)
2. Hightower, J., Borriello, G.: Location Systems for Ubiquitous Computing. Computer 34(8), 57–66 (2001)
3. Navidi, W., Murphy, W.S., Hereman, W.: Statistical methods in surveying by trilateration. Computational Statistics and Data Analysis - Elsevier 27, 209–227 (1998)
4. Walworth, M., Mahajan, A.: 3D Position sensing using the diference of the time-of-flight from a wave source to various receivers. In: ICAR 1997, pp. 611–616 (1997)
5. Hazas, M., Ward, A.: A High Performance Privacy-Oriented Location System. In: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications (PERCom), pp. 216–223 (2003)
6. Yarlagadda, R., Ali, I., Al-Dhahir, N., Hershey, J.: GPS GDOP metric. IEEE Proc.-Radar Sonar Navigation 147, 259–264 (2000)
7. Urruela, A., Sala, J., Riba, J.: Average Performance Analysis of Circular and Hyperbolic Geolocation. IEEE Transactions on Vehicular Technology 55(1), 52–66 (2006)
8. Nishida, Y., Aizawa, H., Hori, T., Hoffman, N.H., Kanade, T., Kakikura, M.: 3D Ultrasonic Tagging System for Observing Human Activity. In: IEEE/RSJ International Conference of Intelligent Robots and Systems (IROS), pp. 1–7 (2003)
9. Yin, M., Shu, J., Liu, L., Zhang, H.: The Influence of Beacon on DV-hop in Wireless Sensor Networks. In: GCCW 2006. Fifth International Conference on Grid and Cooperative Computing Workshops, pp. 459–462 (2006)
10. Ray, P.K., Mahajan, A.: A genetic algorithm-based approach to calculate the optimal configuration of ultrasonic sensors in a 3D position estimation system. Robotics and Autonomous Systems 41, 161–177 (2000)
11. Yang, B., Scheuing, J.: Cramer-Rao bound and optimum sensor array for source localization from time differences of arrival. In: Proceeding IEEE ICASSP, pp. 961–964 (2005)
12. Yang, B., Scheuing, J.: A Theorical Analisys Of 2D Sensor Arrays for TDOA based Localization. In: Proceeding IEEE ICASSP, vol. 4, pp. 901–904 (2006)
13. Laguna, M., Roa, J.O., Jimenez, A.R., Seco, F.: Diversified Local Search for the Optimal Layout of Beacons in an Indoor Positioning System. Submitted for Publication by Colorado Univ. & IAI-CSIC (2007)

Selection Pressure Driven Sliding Window Behavior in Genetic Programming Based Structure Identification*

Stephan Winkler[1], Michael Affenzeller[2], and Stefan Wagner[2]

[1] Upper Austrian University of Applied Sciences
Research Center Hagenberg
[2] Department of Software Engineering
Softwarepark 11, 4232 Hagenberg, Austria
{stephan,michael,stefan}@heuristiclab.com

Abstract. Virtual sensors are a key element in many modern control and diagnosis systems, and their importance is continuously increasing; if there are no appropriate models available, virtual sensor design has to be based on data. Structure identification using Genetic Programming is a method whose ability to produce models of high quality has been shown in many theoretical contributions as well as empirical test reports. One of its most prominent shortcomings is relatively high runtime consumption; additionally, one often has to deal with problems such as overfitting and the selection of optimal models out of a pool of potential models that are able to reproduce the given training data.

In this article we present a sliding window approach that is applicable for Genetic Programming based structure identification; the selection pressure, a value measuring how hard it is to produce better models on the basis of the current population, is used for triggering the sliding window behavior. Furthermore, we demonstrate how this mechanism is able to reduce runtime consumption as well as to help finding even better models with respect to test data not considered by the training algorithm.

1 Genetic Programming Based Structure Identification

Virtual sensors are widely understood as calculation models that can be used instead of physical sensors; they are a key element in many modern control and diagnosis systems, and their importance is continuously increasing. In case no appropriate mathematical models are available (to the required precision), virtual sensor design must be based on data. In most such cases of systems identification, universal approximators (as for example Artificial Neural Networks) are commonly used, even though their limits are well known. Unlike these methods, Genetic Programming (GP) based techniques have been used successfully

* The work described in this paper was done within the Translational Research Project L282 "GP-Based Techniques for the Design of Virtual Sensors" sponsored by the Austrian Science Fund (FWF).

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 788–795, 2007.
© Springer-Verlag Berlin Heidelberg 2007

for solving data based structure identification problems in various different areas of data mining. Within the last years we have set up a further developed, fully automated and problem domain independent GP based structure identification framework that has been successfully used in the context of various different kinds of identification problems for example in mechatronics ([2], [10]), medical data analysis [8] and the analysis of steel production processes [9].

Basically, GP is based on the theory of Genetic Algorithms (GAs). A GA works with a set (population) of solution candidates. New individuals are created on the one hand by combining the genetic make-up of two previously selected solution candidates (by "crossover"), and on the other hand by mutating some individuals, which means that randomly chosen parts of genetic information are changed (which is normally applied on a minor ratio of the algorithm's population). Each individual is evaluated using a pre-defined fitness function.

Similar to GAs, GP also utilizes a population of solution candidates which evolves through many generations towards a solution using certain evolutionary operators and a selection scheme increasing better solutions' probability of passing on genetic information. The main difference is that, whereas GAs are intended to find an array of characters or integers representing the solution of a given problem, the goal of a GP process is to produce a computer program solving the optimization problem at hand. In the case of structure identification, solution candidates represent mathematical models (for example stored as formula structure trees). These are evaluated by applying the formulae to the given training data and comparing the so generated output to the original target data. Typically, the population of a GP algorithm contains a few hundred individuals and evolves through the action of operators known as crossover, mutation and selection. The left part of Fig. 1 visualizes how the GP cycle works: As in every evolutionary process, new individuals (in GP's case, new programs) are created and tested, and the fitter ones in the population succeed in creating children of their own; unfit ones die and are removed from the population [5].

As already stated before, the results achieved for various data based identification problems using GP are satisfying, but still there are several problems that are to be considered, the most obvious ones being on the one hand the rather high runtime consumption and on the other hand over-fitting (which means that the algorithm produces models that show a good fit on training data but perform badly on test data). Even though the GP-based approach has been shown to be much less likely to over-fit [8], still this is an issue that is always present in the context of data based modeling.

2 On-Line and Sliding Window Genetic Programming

The idea of sliding window behavior in computer science is not novel; in machine learning, drifting concepts are often handled by moving the scope (of either fixed or adaptive size) over the training data (see for example [7] or [3]). The main idea is the following: Instead of considering all training data for training models (in the case of GP, for evaluating the models produced), the algorithm initially

only considers a part of the data. Then, after executing learning routines on the basis of this part of the data, the range of samples under consideration is shifted by a certain offset. Thus, the window of samples considered is moved, it slides along the training data; this is why we are talking about sliding window behavior.

When it comes to GP based structure identification, sliding window approaches are not all too common; in general, the method is seen as a global optimization method working on a set of training samples (which are completely considered by the algorithm within the evaluation of solution candidates). On the contrary, GP is often even considered as an explicitely off-line, global optimization technique. Nevertheless, during research activities in the field of on-line systems identification [10], we discovered several surprising aspects. In general, on-line GP was able to identify models describing a Diesel engine's NO_x emissions remarkably fast; the even more astonishing fact was that these models were even less prone to over-fitting than those created using standard methods. After further test series and reconsidering the basic algorithmic processes, these facts did not seem to be surprising to us anymore: On the one hand, especially the fact that the environment, i.e. the training data currently considered by the algorithm, is not constant but rather changing during the execution of the training process, contributes positively to the models' quality, it obviously decreases the threat of overfitting. On the other hand, the interplay of a changing data basis and models created using different data also seems to be contributing in a positive way. As the on-line algorithm is executed and evaluates models using (new) current training data forgetting samples that were recorded in the beginning, those "old" data are really forgotten from the algorithm's point of view. Still, the models created on the basis of these old data are still present. The behavior that results out of this procedure is more or less that several possible models that explain the first part of the data are created, and as the scope is moved during the algorithm's execution, only those models are successful that are also able to explain "new" training data.

So, the most self-evident conclusion was that these benefits of online training should be transferred to off-line training using GP. Obviously, this directly leads us to sliding window techniques which are described in the following sections.

3 Selection Pressure as Window Moving Trigger

One of the most important problem independent concepts used in our implementation of GP-based structure identification is Offspring Selection [1], an enhanced selection model that has enabled Genetic Algorithms and Genetic Programming implementations to produce superior results for various kinds of optimization problems[1]. As in the case of conventional GAs or GP, offspring are generated by parent selection, crossover, and mutation. In a second (offspring) selection

[1] In fact, this procedure of eliminating offspring that do not meet the given requirements (i.e. that are assigned a quality value worse than their parents) is also integrated in the left part of Figure 1.

step (as it is used in our GP implementation), only those children become members of the next generation's population that outperform their own parents, all other ones are discarded. The algorithm therefore repeats the process of creating new children until the number of successful offspring is sufficient to create the next generation's population[2]. Within this selection model, selection pressure is defined as the ratio of generated candidates to the population size:

$$SelectionPressure = \frac{\mid Solution\ \ candidates\ \ created \mid}{\mid Successful\ \ solution\ \ candidates \mid}$$

The higher this values becomes, the more models have to be created and evaluated in order to produce enough models that are supposed to form the next generation's population. In other words, this selection pressure is a value giving a measure of how hard it is for the algorithm to produce a sufficient number of successful solution candidates. This simplified Offspring Selection model is schematically displayed in the right part of Figure 1.

The proposed idea is to initially reduce the amount of data that is available for the algorithm as identification data. As the identification process is executed, better and better models are created which leads to a rise of the selection pressure; as soon as the selection pressure reaches a predefined maximum value, the limits of the identification data are shifted and the algorithm goes on considering another part of the available identification data set. This procedure is then repeated until the actual training data scope has reached the end of the training data set available for the identification algorithm, i.e. when all data have been considered. By doing so, the algorithm is (following the considerations formulated in the previous section) even less exposed to over-fitting, and due to the fact that the models created are evaluated on much smaller data sets we also expect a significant decrease of runtime consumption.

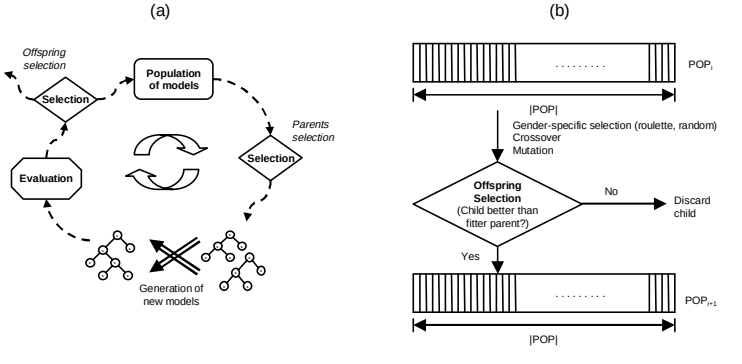


Fig. 1. (a) The Extended Genetic Programming Cycle including Offspring Selection; (b) Embedding a simplified version of Offspring Selection into the GP process

[2] In fact, there are several aspects of Offspring Selection that are not explained here; a more detailed description can be found in [1].

Algorithm 1. The sliding window based GP structure identification process.

function Model = SlidingWindowGPStructId (TrainingData, FunctionalBasis, WindowStartSize, WindowStepSize, MaximumWindowSize, MaximumSelectionPressure1, MaximumSelectionPressure2)
  $index1 = 1$, $index2 = WindowStartSize - 1$
  $InitializeModelsPool$
  while $index2 <= Data.Length$ **do**
    while $CurrentSelectionPressure <= MaximumSelectionPressure1$ **do**
      Perform GP-based structure identification using the given $TrainingData$ in the interval $[index1; index2]$
    end while
    $index2 = Min((index2 + WindowStepWidth), Data.Length)$
    $index1 = Max((index2 - MaximumWindowSize + 1), 0)$
  end while
  $index2 = Data.Length$
  $index1 = Max((Data.Length - MaximumWindowSize + 1), 0)$
  while $CurrentSelectionPressure <= MaximumSelectionPressure2$ **do**
    Perform GP-based structure identification using the given $TrainingData$ in the interval $[index1; index2]$
  end while
return Current best model

In Algorithm 1 we give a sketch of the sliding window GP based structure identification process incorporating Offspring Selection. The standard GP parameters (as, for example, population size, mutation rate and crossover operator combinations) are hereby omitted; we only describe the sliding window specific process modifications. The $WindowStartSize$ parameter gives the initial size of the current data window; as soon as the current selection pressure reaches $MaximumSelectionPressure1$, the window is moved by $WindowStepSize$ samples; the $MaximumWindowSize$ parameter specifies the maximum size of the current training data scope. This procedure is repeated until the end of the data set is reached; in the end, the process stops as soon as the second maximum selection pressure parameter value is reached (which does not necessarily have to be the same as the first maximum selection pressure value).

4 Experiments

4.1 Experimental Setup, Data Basis

For testing the sliding window approach described here we have used the *HeuristicModeler*, a GP-based data mining approach implemented within the HeuristicLab [6], a paradigm-independent and extensible environment for heuristic optimization. In the following we will report on tests executed using the *Thyroid* data set, a widely used machine learning benchmark data set containing the results of medical measurements which were recorded while investigating patients

potentially suffering from hypotiroidism[3]. In short, the task is to determine whether a patient is hypothyroid or not; three classes are formed: normal (not hypothyroid), hyperfunction and subnormal functioning. In the following we are going to report on test results achieved using the first 80% of the data (containing 7,200 samples in total) as training data and the remaining 20% for testing the models created. We here state the quality of the classifiers created by the identification process using the mean squared error function for evaluating them.

4.2 Parameter Settings and Test Results

For testing the sliding window approach presented in this paper and also for comparing its ability to produce models of high quality we have tested the following 5 different GP-based data mining strategies characterized by their population size $|pop|$, maximum selection pressure values MSP (maximum selection pressure, $MSP1$ and $MSP2$ (maximum selction pressure values 1 and 2 as explained in the previous section) and relative values for sliding window parameters:

1. Standard-GP: $|pop| = 2000$, 1500 generations, no Offspring Selection.
2. GP including Offspring Selection: $|pop| = 1000$, $MSP = 200$
3. Sliding window GP: $|pop| = 1000$, $MSP1 = 50$, $MSP2 = 200$, sliding window: initial size 0.2, step width 0.1, maximum size 0.4
4. Sliding window GP: $|pop| = 1000$, $MSP1 = 50$, $MSP2 = 200$, sliding window: initial size 0.4, step width 0.2, maximum size 0.5
5. Sliding window GP: $|pop| = 1000$, $MSP1 = 20$, $MSP2 = 200$, sliding window: initial size 0.2, step width 0.05, maximum size 0.4

All tests were executed applying 15% mutation rate and a combination of random and roulette parent selection schemata. For each test scenario we have executed 5 independent test runs. In the following table we give average numbers of iterations and solutions evaluated as well as the quality of the models identified (average values as well as the quality of each test series' model showing the best fit on the complete training data set) with respect to (complete) training and test data. Please note that all variables were normalized independently (i.e. scaled linearily so that the resulting variables' mean values are equal to 0 and their standard deviations are exactly 1.0). The maximum size of models created by the training algorithm was set to 60, the maximum formula tree height to 8.

In Figure 2 we give two characteristic screenshots: In the left part the selection pressure progress of one of the test runs of test series (5) is displayed (with vertical gray lines indicating training data scope drifts: every time the selection pressure became greater than 20, the window was shifted); in the right part we show a graphical representation of the evaluation of the best classifier trained in test series (4). This result is indeed remarkable as it correctly classifies 98.46% of the training and 98.08% of the test samples. This model even outperforms those reported on in [8]; a detailed confusion matrix is given in Table 2.

[3] Further information about the data set used can be found on the UCI homepage `http://www.ics.uci.edu/~mlearn/`.

Table 1. Results of the tests executed for the *Thyroid* data set

Test Scenario	Iterations (Average)	Solutions Evaluated (Average)	Speed Up	Model Quality on Training Data		on Test Data	
				Best Model	Average	Best Model	Average
1	1,500.00	3,000,000.00	1.00	0.316	0.410	0.381	0.444
2	64.40	2,717,678.80	1.10	0.155	0.283	0.251	0.341
3	65.60	3,199,551.00	2.40	0.166	0.193	0.219	0.233
4	59.80	1,925,755.40	3.18	0.166	0.246	0.199	0.310
5	62.60	2,483,116.60	3.10	0.125	0.173	0.220	0.252

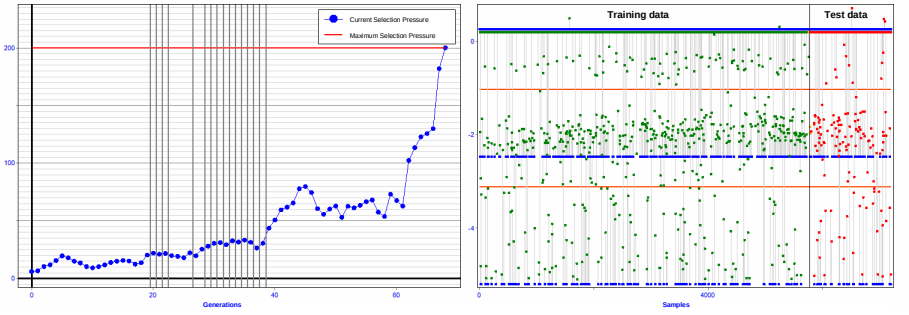


Fig. 2. Left: Selection pressure progress of the best test run of test series (5); Right: Graphical representation of the best test run of test series (4)

Obviously, as is summarized in Table 1, all GP methods using Offspring Selection perform significantly better that the standard implementation. Furthermore, the use of sliding window mechanisms here resulted in models that perform better on test data as well as in significant runtime reduction. Due to the fact that on the one hand not all training data but only the respective current data scopes are used in the sliding window test series and on the other hand the share of model evaluation in runtime consumption of GP based data mining is almost 100%, the algorithms are executed significantly faster: The respective speed up values range from 2.4 to almost 3.2.

Table 2. Analysis of the best model produced in test series (4) whose evaluation is displayed in Figure 2

Original Class →		1	2	3	
Class Predicted	1	29 (2.06%)	0 (0.00%)	2 (0.14%)	
	2	5 (0.35%)	63 (4.47%)	19 (1.35%)	
	3	0 (0.00%)	1 (0.07%)	1290 (91.55%)	
Correctly Classified					1382 (98.08%)

5 Discussion, Summary

In this article we have presented a sliding window approach for data mining using evolutionary computation techniques, namely enhanced Genetic Programming. A further developed selection model (Offspring Selection), is used for determining the resulting selection pressure which is used for triggering the drift of the current training data scope. In the experimental part we have reported on a series of tests using a widely used classification benchmark problem for demonstrating the effects of the use of these enhanced aspects. It has been shown that it is possible to reduce the algorithm's runtime as well as to increase the models' test quality when applying the sliding window mechanism presented here.

Additional to further test series, more detailed analysis of this method is still needed. For example, the effects of this drifting mechanism on the genetic diversity are to be analyzed; we will report on this analysis in [11].

References

1. Affenzeller, M., Wagner, S.: Offspring Selection: A New Self-Adaptive Selection Scheme for Genetic Algorithms. Adaptive and Natural Computing Algorithms, pp. 218–221 (2005)
2. Alberer, D., Del Re, L., Winkler, S., Langthaler, P.: Virtual Sensor Design of Particulate and Nitric Oxide Emissions in a DI Diesel Engine. In: Proceedings of the 7thInternational Conference on Engines for Automobile ICE (2005) paper nr. 2005-24-063 (2005)
3. Hulten, G., Spencer, L., Domingos, P.: Mining Time-Changing Data Streams. In: Proceedings of the 7^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 97–106. ACM Press, New York (2001)
4. Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge (1992)
5. Langdon, W., Poli, R.: Foundations of Genetic Programming. Springer, New York (2002)
6. Wagner, S., Affenzeller, S.: Heuristiclab: A Generic and Extensible Optimization Environment. Adaptive and Natural Computing Algorithms, 538–541 (2005)
7. Widmer, G., Kubat, M.: Learning in the Presence of Concept Drift and Hidden Contexts. Machine Learning 23(2), 69–101 (1996)
8. Winkler, S., Affenzeller, M., Wagner, S.: Using Enhanced Genetic Programming Techniques for Evolving Classifiers in the Context of Medical Diagnosis - An Empirical Study. GECCO 2006 Workshop on Medical Applications of Genetic and Evolutionary Computation (MedGEC 2006), paper nr. WKSP115. ACM Press, New York (2006)
9. Winkler, S., Efendic, H., Del Re, L.: Quality Pre-Assesment in Steel Industry Using Data Based Estimators. In: Cierpisz, S., Miskiewicz, K., Heyduk, A. (eds.) Proceedings of the MMM 2006 Workshop on Automation in Mining, Mineral and Metal Industry. International Federation for Automatic Control, pp. 185–190 (2006)
10. Winkler, S., Efendic, H., Affenzeller, M., Del Re, L., Wagner, S.: On-Line Modeling Based on Genetic Programming. Int. J. on Intelligent Systems Technologies and Applications, 2(2/3), 255–270. Inderscience Publishers (2007)
11. Winkler, S., Affenzeller, M., Wagner, S.: Genetic Diversity in Systems Identification Based on Extended Genetic Programming. In: Accepted to be published in Proceedings of the 16^{th} International Conference on Systems Science (2007)

Multi-Objective Evolutionary Algorithms Using the Working Point and the TOPSIS Method

Máximo Méndez[1] and Blas Galván[2]

[1] Computer Science Department, University of Las Palmas de Gran Canaria Edif. de Informática y Matemáticas. Campus de Tafira, CP 35017 Las Palmas, Spain
mmendez@dis.ulpgc.es
[2] IUSIANI, University of Las Palmas de Gran Canaria Edif. Polivalente. Campus de Tafira, CP 35017 Las Palmas, Spain
bgalvan@step.es

Abstract. The use of Multi-Ojective Evolutionary Algorithm (MOEA) methodologies, distinguished for its aptitude to obtain a representative Pareto optimal front, cannot always be the most appropriate. In fact, there exist multi-objective engineering problems that identify one feasible solution in the objective space known as Working Point (WP), not necessarily Pareto optimal. In this case, a Decision Maker (DM) can be more interested in a small number of solutions, for example, those that located in a certain region of the Pareto optimal set (the WP-region) dominate the WP. In this paper, we propose WP-TOPSISGA, an algorithm which merges the WP, MOEA techniques and the Multiple Criteria Decision Making (MCDM) method TOPSIS. With TOPSIS, a DM only needs input the preferences or weights w_i, with our method, however, the weights are evaluated by interpolation in every iteration of the algorithm. The idea is to guide the search of solutions towards the WP-region, giving an order to the found solutions in terms of Similarity to the Ideal Solution.

Keywords: Multi-Objective Optimization, Preferences, Working Point, Decision Making, TOPSIS, Evolutionary Algorithms.

1 Introduction

There exist multi-objective engineering problems that identify one stable and feasible solution in the objective space known as Working Point (WP), not necessarily effective or Pareto optimal. In these types of problems the use of Multi-Objective Evolutionary Algorithms (MOEAs), characterized by its aptitude to find wide and well distributed solution fronts, as a previous step before a Decision Making process, does not always turn out the most appropriate procedure. In fact, a Decision Maker (DM) plunged in the task of improving the WP, can be more interested in a small number of solutions, for example those that located in a certain region of the Pareto optimal set (the WP-region) dominate the WP,

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 796–803, 2007.
© Springer-Verlag Berlin Heidelberg 2007

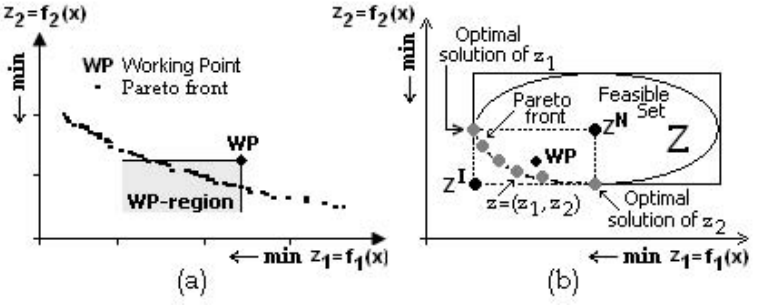


Fig. 1. (a) The WP-region. (b) The objective space.

see Fig. 1(a). The incorporation of preferences and reference points in MOEAs is not new [1],[3],[5], though we do not know previous works that incorporate the WP as reference point. In this work, we propose WP-TOPISISGA, an algorithm which combines the WP, the second generation of MOEAs (we use NSGA-II [2]) and the Multiple Criteria Decision Making (MCDM) method TOPSIS [4]. With TOPSIS, a DM only needs input the preferences or weights w_i, with our method, however, the weights are evaluated by interpolation in every iteration of the algorithm. The idea is to lead the search of solutions towards the region of interest (the WP-region) and to induce an order in accordance with the Similarity to the Ideal Solution. In order to illustrate the advantages of the proposed algorithm, the binary multi-objective knapsack problem (0-1 MOKP) has been used as a numerical example.

2 Multi-objective Problem Description

In general, a multi-objective problem can be defined as follows:

$$\begin{cases} min. \quad f_1(x), f_2(x), ..., f_k(x) \\ \quad s.t. \\ \quad\quad g_j(x) \leq 0 \qquad j \in (1, 2, ..., l) \\ \quad\quad h_j(x) = 0 \qquad j \in (1, 2, ..., m) \end{cases} \tag{1}$$

where k$\geq$ 2 represents the number of objective functions, x=$(x_1, x_2, ..., x_n) \in X$ is the solution vector of n decision variables and X is the set of realizable solutions in the decision space. To represent the set of realizable solutions in the objective space, it is necessary to determine the image of every solution of the decision space. We obtain thus a set Z, see Fig. 1(b), defined as follows:

$$Z = \{z = (z_1, ..., z_k) \in R^k : z_1 = f_1(x), ..., z_k = f_k(x), \ \forall x \in X\} \tag{2}$$

here, the set Z represents the realizable solutions in the objective space and z=$(z_1, ..., z_k)$ a solution of the objective space.

Definition 1. *A solution* $z^u = (z_1^u, ..., z_k^u)$ *dominates a solution* $z^t = (z_1^t, ..., z_k^t)$ *if and only if:* $\forall i \in (1, ..., k)\ z_i^u \leq z_i^t$ *and* $\exists i \in (1, ..., k) : z_i^u < z_i^t$.

Definition 2. *A solution* $x^* \in X$ *is Pareto optimal if and only if there not exist a solution* $x \in X$ *such as* $x \in X$ *dominates* $x^* \in X$. *A Pareto optimal solution is also called non-dominated or effective.*

Definition 3. *A set of Pareto optimal solutions is said to be a Pareto set or an effective set.*

Definition 4. *A set of vectors in the objective space that are image of a Pareto set is said to be a Pareto front.*

Definition 5. *The ideal solution* $z^I = (z_1^I, z_2^I, ..., z_k^I)$ *is the vector which is constructed with the best objective function values of all objective space solutions, see Fig. 1(b). Generally, this solution is unfeasible.*

Definition 6. *The nadir solution* $z^N = (z_1^N, z_2^N, ..., z_k^N)$ *is the vector which is constructed with the worst objective function values of all Pareto front solutions, see Fig. 1(b). This solution can be feasible or unfeasible.*

Definition 7. *To be a multi-objective engineering problem, the Working Point WP=*$z^{WP} = (z_1^{WP}, ..., z_k^{WP})$ *is defined, in terms of objective space solutions, as that solution which identifies a stable solution of functioning, not always effective or Pareto optimal, see Fig. 1(b). The WP can also be defined as that solution which defines the current state of the system.*

Definition 8. *To be a multi-objective engineering problem, the WP-region is defined as that subset of solutions of Pareto front which dominates the Working Point, see Fig. 1(a).*

Axiom 1. If a solution $z^{WP} = (z_1^{WP}, ..., z_k^{WP})$ is the Working Point of a multi-objective engineering problem, then $z^{WP} = (z_1^{WP}, ..., z_k^{WP})$ belongs to the feasible set of solutions in the objectives space Z.

3 TOPSIS Method

The TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution) method was developed by Hwang and Yoon [4] for solving MCDM problems with a finite number of solutions. The TOPSIS method minimizes the distance to the ideal solution while maximizing the distance to the nadir. The distances are calculated with a particular value of p ($1 \leq p \leq \infty$) of the Minkowski's metrics $d_p = \left\{ \sum_{i=1}^{k} w_i^p \, |z_i - z_i^*|^p \right\}^p$, where z_i^* ($i \in \{1, 2, ..., k\}$) is a reference solution and w_i are the weights assigned by the DM. The TOPSIS procedure consists of:

(1) Obtain a decision matrix, where a set of alternatives (solutions) A=(a_j, j=1,2,...,n) is compared with respect to a set of criterion functions (objective functions) C=(c_i, i=1,2,...,k). An element x_{ij} of the matrix, is a value indicating the performance rating of nth alternative with regard to the criterion c_i.
(2) Convert raw values x_{ij} into normalized values r_{ij}.
(3) Calculate the weighted normalized values as: $v_{ij} = w_i r_{ij}$
where w_i is the weight of the ith criterion set by the DM and $\sum_{i=1}^{n} w_i = 1$.
(4) Identifie the ideal solution $v_i^I = (v_1^I, ..., v_k^I)$ and the nadir $v_i^N = (v_1^N, ..., v_k^N)$.
(5) Determinate the Euclidean distances (d_j^I, d_j^N) for each solution from the ideal solution and from the nadir.
(6) For each solution, calculate the relative closeness to the ideal solution (rating of Similarity to the Ideal Solution) as:

$$D_j^I = \frac{d_j^N}{d_j^I + d_j^N} \tag{3}$$

(7) Sort the solutions in terms of Similarity D_j^I, from the most Similarity to the least.

4 Proposed Method

The proposed algorithm WP-TOPSISGA (the flow chart is shown in Fig. 2) can be adapted to any MOEA and uses basically the original formulation of NSGA-II [2], but the crowding operator has been replaced by the "Similarity operator"; moreover the sizes of the population P_A and the archive of non-dominated solutions P_A^t can differ (we use a size M for the population and N for the archive). Initially, M individuals are randomly generated and the archive of non-dominated solutions is set empty. In each generation, the solutions from both the population and the archive are combined, classified in fronts of non-domination and assigned to the archive P_A^{t+1}. In most of the cases, the number of non-dominated solutions found could be greater than the archive size (N). In order to set the archive size to N solutions, an operator (we call it "Similarity operator") is used in the following form: the extremes of the first front of range 1 define the ideal and nadir solutions, with this information the Similarity to the Ideal Solution D_j^I is computed, sorted in ascending order and used to choose a subset of solutions from the last front which cannot be entirely chosen, but the nearest solution to WP and the front extremes have the maximum priority. Afterwards a reproductive selection (the criterion is: (i) smaller non-dominated rank, (ii) bigger Similarity) of individuals ramdomly selected from P_A^{t+1} is accomplished using a binary tournament and a mating pool (MP) is filled up, at this stage M new individuals are generated by applying recombination operators on MP.

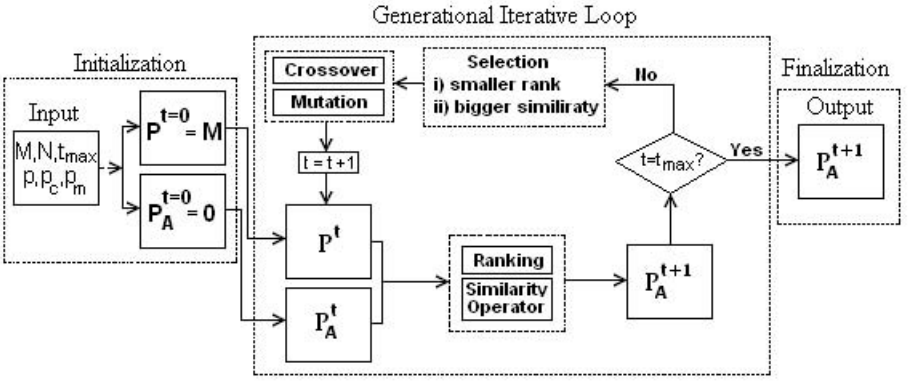


Fig. 2. Flow chart of WP-TOPSISGA

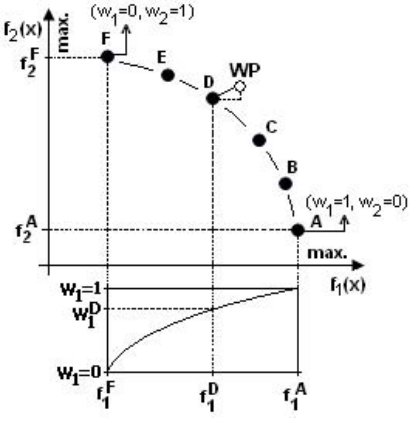


Fig. 3. Estimation of the weights w_i

It is important to mention that in each MOEA iteration, the preferences or weight values w_i are required to compute D_j^I. Now we suppose a certain generation t of the search and an optimal resultant front (Fig. 3 in terms of maximisation). The election for a DM of the extreme solution A, would indicate that the objective f_1 is of maximum importance compared with the objective f_2 (in quantitative values of the preferences: w_1=1, w_2=0). In contrary, if the election of the DM was the extreme solution F, now it would be f_2 compared with f_1, the objective of maximum importance (in quantitative values of the preferences: w_1=0, w_2=1). If now the choice of the DM was the solution D (solution the nearest to the WP), the value of w_1 would be estimated by interpolation (in addition: w_2=1-w_1). With this principle, the proposed method estimates the weights w_i by interpolation from the nearest solution to WP.

5 Computational Experiments for the 0-1 MOKP

We here illustrate our method WP-TOPSISGA by applying it, to the binary multi-objective knapsack problem (0-1 MOKP). In all simulations, the population was set to M=200 individuals, the crossover probability to p_c=0.8, the mutation rate to p_m=0.01, the p value to 2 and the archive size of non-dominated solutions was changed progressively following the sequence N=10,20,30,40,50,100 individuals. The maximum generation number was G=500.

5.1 The 0-1 MOKP Description

The 0-1 MOKP problem is well known in multi-objective optimization and can be considered as a general enunciate of any integer programming problem with binary variables and positive coefficients. The problem has complexity NP-Hard and models numerous real applications in different fields of the Economy, the Finance and in general, in any problem that adjusts to the model described in (4). Various evolutionary algorithms have been utilized to solve the 0-1 MOKP [5],[7].

In general, in the knapsack problem, n knapsacks with its corresponding capacities b_i are considered as well as a set of m objects with different weights w_{ij} and profits c_{ij}. The problem is to find a subset of items which maximizes the total of the profits in the subset, but considering the constraints of capacity (maximum weight) of each knapsack. The 0-1 MOKP can be formulated as:

$$\begin{cases} max. & f_i(x) = \sum_{j=1}^{m} c_{ij}x_j & i = 1, 2, ..., n \\ Such\ that & \sum_{j=1}^{m} w_{ij}x_j \leq b_i & x_j \epsilon \{0, 1\} \end{cases} \tag{4}$$

where:

m = number of items — c_{ij} = profit of item j according to knapsack i
$\mathbf{x}_j$ = a decision variable — w_{ij} = weight of item j according to knapsack i
n = number of knapsacks — b_i = capacity of knapsack i

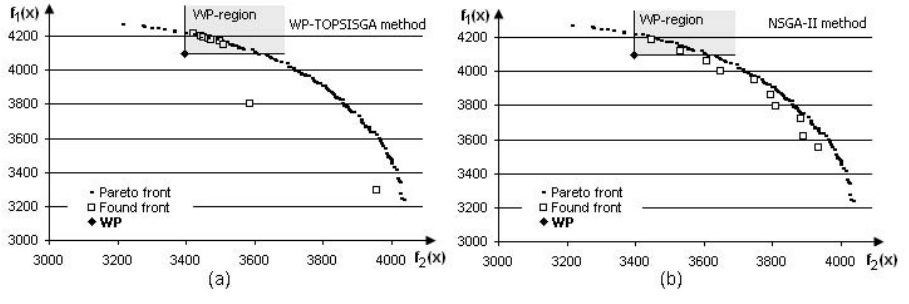


Fig. 4. Non-dominated front found by: (a) WP-TOPSISGA, (b) NSGAII

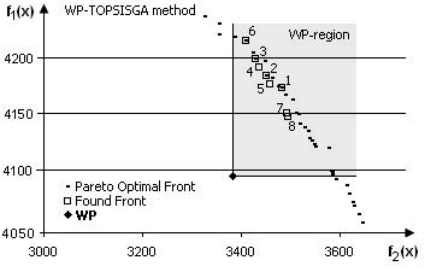


Fig. 5. Pre-order in non-dominated front found by WP-TOPSISGA

Table 1. Metric C(R,A) values for 500 generations

Method	N=10	N=20	N=30	N=40	N=50	N=100
WP-TOPSISGA	57.62	63.11	60.26	80.49	85.57	96.66
NSGA-II	100	96.66	90.95	92.14	91.61	95.25

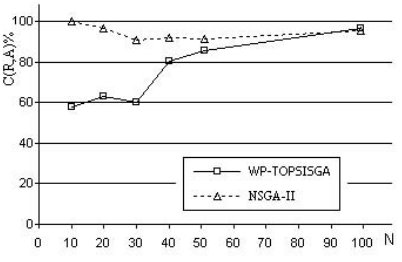


Fig. 6. Graphic view of table 1

Two knapsacks and 100 items have been used as data. The true Pareto front is known (121 solutions).

Fig. 4(a) and Fig. 4(b) show the Pareto front determined by WP-TOPSISGA and NSGA-II respectively with N=10 and WP=(f_1=4097, f_2=3395). From these figures, we note that WP-TOPSISGA conducts the search towards the region of interest (the WP-region) while NSGA-II finds an even final set. Fig. 5 zoomed in on the WP-region of Fig. 4(a). From it, we notice that there is, in the found front, a pre-order in accordance with the Similarity to the Ideal Solution.

Finally, WP-TOPSISGA was compared with NSGA-II based on the C metric [6], using the efficient frontier of the WP-region as a reference set R. Table 1 reports the percentage (average after ten runs) of the final outcomes (labelled A) dominated by the set R and it shows that the proportion of efficient frontier reached by the algorithms is larger when we use WP-TOPSISGA and when the size of the non-dominated population (N) is small, but this difference seems to disappear when N increases. Fig. 6 shows graphically the results of table 1.

6 Conclusions

In this paper, we have presented WP-TOPSISGA, an algorithm which merges the Working Point, MOEA techniques and the MCDM method TOPSIS. The idea is to lead the search towards that subset of solutions of Pareto front (the WP-region) that dominates the Working Point. With this intention, a method of evaluation of the preferences or weights w_i based on interpolation is applied. The realized experiments reveal that WP-TOPSISGA concentrates the found solutions in the WP-region and classifies the solutions in terms of Similarity to the Ideal Solution. Besides, the proposed approach was compared with NSGA-II based on the C metric, using the efficient frontier of the WP-region as a reference set R, and shows that for small population sizes of non-dominated solutions, the percentage of solutions of the Pareto optimal set reached is glaringly better with WP-TOPSISGA than NSGA-II, though this difference disappears for major values in the population size of non-dominated solutions. Nevertheless, the different conceptions of both algorithms must be emphasized, WP-TOPSISGA concentrates and arranges the set of resultant solutions in the WP-region while NSGA-II obtains a wide and well distributed set of solutions.

References

1. Cvetkovic, D., Coello, C.: Human Preferences and Their Applications in Evolutionary Multi-Objective Optimization. In: Yaochu, J. (ed.) Knowledge Incorporation in Evolutionary Computation (Studies in Fuzziness and Soft Computing, vol. 167, pp. 479–502. Springer, Heidelberg (2005)
2. Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
3. Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S.: Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. International Journal of Computational Intelligence Research 2(3), 273–286 (2006)
4. Hwang, C.L., Yoon, K.: Multiple Attribute Decision Making: Methods and applications. Springer, Heidelberg (1981)
5. Méndez, M., Galván, B., Salazar, D., Greiner, D.: Multiple-Objective Genetic Algorithm Using The Multiple Criteria Decision Making Method TOPSIS. Presented at The 7th International Conference devoted to Multi-Objective Programming and Goal Programming 2006 (MOPGP 2006), Tours, France (June 2006)
6. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary Computation Journal 8(N2), 125–148 (2000)
7. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, Swiss Federal Institute of Technology Zurich (1999)

Self-organizing Feature Maps to Solve the Undirected Rural Postman Problem

M.L. Pérez-Delgado and J.C. Matos-Franco

Universidad de Salamanca, Av. Requejo, 33, C.P. 49022, Zamora, Spain
mlperez@usal.es, jcmatos@usal.es

Abstract. The Rural Postman Problem consists of finding a shortest tour containing all edges in a subset, in the subgraph induced by some subset of nodes. In general, the problem is NP-hard, since the Traveling Salesman Problem can be easily transformed into it. The Traveling Salesman Problem consist of finding a shortest closed tour which visits all the cities in a given set. Artificial Neural Networks have been applied to solve the Traveling Salesman Problem in the last years. In this work we propose to apply self-organizing feature maps to solve the first problem, transforming it previously into the second.

Keywords: Rural Postman Problem, Traveling Salesman Problem, Self-organizing Feature Maps.

1 Introduction

The *Rural Postman Problem* (RPP) was first defined by Orloff [1]. Let $G = (V, E, c)$ be an undirected connected graph, where V is a set of points, $E = \{(i, j) : i, j \in V\}$ is a set of edges and c is a function that associates a nonnegative cost to each edge. Let F be a subset of E. The RPP consists of determining a minimum cost tour of G such that each edge in F is traversed at least once. The problem appears in a variety of practical contexts like control of plotting and drilling machines, optimization of laser-plotter beam movements, mail delivery, network maintenance, garbage collection, check of the links of a web site, street cleaning or school bus routing [2], [3], [4], [5].

The RPP is NP-hard [5], since the *Traveling Salesman Problem* (TSP) can easily be transformed into it [6]. Let $G = (V, E, d)$ be a graph, where V is the set of points or cities in the problem, E is the set of connections between pairs of cities and d is a function that associates a cost to each connection in the graph. The objective of the TSP is to find a closed path of minimum length that visits each city once. The TSP is a NP-hard combinatorial optimization problem. Therefore, heuristic techniques are applied to reduce the search space [7], [8]. The artificial neural networks take a prominent place among these techniques.

There are exact algorithms to solve the RPP [9], [10], [11], [12]. Because the RPP is a NP-hard problem, it has been tackled with some heuristics, due to the difficulty of using exact approaches to global optimality [13], [14], [15].

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 804–811, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In this paper we present a new heuristic solution based on the application of *self-organizing feature maps* (SOFM). There are no many works about the application of neural models to solve the RPP. Because the SOFM works with Euclidean distances, we will consider an undirected RPP defined on the Euclidean plane, that has some practical applications [3], [4]. Let us consider the problem of drawing with a plotter. The figure to be drawn is made up of lines connecting points in the plane. So, the problem can be modeled as an undirected RPP: we consider a graph $G = (V, E, c)$ where V includes all the points in the drawing area, E is the set of lines connecting pairs of points of V, and the function c associates to every line in E the Euclidean distance between their corresponding end points. To draw a figure, we consider a subset of lines, $F \subset E$, that is the set of required lines to draw the figure.

This paper is organized as follows. Section 2 describes the SOFM neural network. In Section 3 we describe the solution proposed to the RPP. Computational results are reported in Section 4, followed by a conclusion.

2 Self-organizing Feature Maps

The SOFM is a neural network model with unsupervised learning proposed by Teuvo Kohonen [16]. This network constructs a topology-preserving mapping from a high-dimensional data space to a low-dimensional space, in such a way that relative distances between data points are preserved.

The neurons in the network are organized in an architecture with two layers. The first one, called input layer, has the same dimension as the input vector. The second one, called competitive layer or map, includes N neurons usually organized as a two-dimensional regular lattice. Every neuron in the input layer, j, is connected to every neuron in the competitive layer, i. Each connection has a weight associated with it, w_{ij}. When an input is presented to the network, it calculates the output vector, that includes N components. The value of each component is calculated as the weighted sum of the inputs and the weights.

The network has two operating modes. In the *running mode* of the network, the weights remain fixed, and when an input vector is presented to the network the output vector is calculated. In the *learning mode* the training patterns are presented iteratively to the network, and the weights of the output neurons are adapted. When an input pattern, x, is presented to the network, the nearest neuron in terms of a certain distance measure is selected as the winner neuron, g. The update of the weights will be applied as well to the winner as to its neighbors, according to the expression:

$$\triangle w_i = \alpha(t)h(|i - g|, R(t))(x - w_i), i = 1, \ldots, N. \tag{1}$$

where h is the neighborhood function, $\alpha(t)$ is a parameter called learning rate, that determines the portion of weight needed to be adjusted, and $(x - w_i)$ is the distance between the input vector and the vector of weights of the neuron i. Various distance measures can be applied (the Euclidean distance, the Minkowski

metric, ...), taking into account that the distance measure must be compatible with the weight update rule of the SOFM.

The neighbor neurons of the winner will be identified by the neighborhood function. This allows that similar input patterns tend to produce a response in neurons that are close to each other in the output layer. The function depends on the current neighborhood radius, $R(t)$, that determines the size of the neighborhood, and the distance from a neuron, i, to the winner, g, $|i - g|$. This distance will be equal to de difference between the positions of i and g in the map. To make possible the correct spread of the map, the neighborhood radius and the learning rate must diminish slowly with the time. For the winner neuron and for their neighbors, the neighborhood function will be different from 0. For the rest, that function will be 0, so then the weights remain unchanged. Some common neighborhood functions are the rectangular, pipe, triangular, exponential-concave and Gaussian functions.

This neural network has been applied in clustering, pattern recognition, speech analysis, industrial and medical diagnostics, exploratory data analysis, robotics, instrumentation and control and to solve combinatorial optimization problems, like the well-known Traveling Salesman Problem [7]. Many variants of the basic algorithm have been proposed (see [7] for an overview).

In the solution proposed by Kohonen, the TSP is defined by a set of N cities with coordinates in the Euclidean plane. It uses a network whose input layer includes two neurons, which receive the two inputs of the network, and whose competitive layer is composed of a ring of m neurons ($m \geq N$). The set of cities is presented in an iterative way, so as the weights of the neurons come near to the coordinates of the cities. To determine the path to follow, we will go from the city associated to neuron i to the one associated to the $i + 1$, for $i = 1, 2, \ldots, m - 1$, going through all the ring of neurons. To close the path, the last section will be given by the segment that links the city associated with the neuron m with the one associated with the neuron 1.

3 The Solution Proposed for the Problem

To solve the RPP by applying SOFM we first get a TSP from the RPP. After applying a SOFM algorithm to that TSP, we obtain a solution of the RPP from the solution of the TSP.

3.1 The Traveling Salesman Problem to be Solved

Let $G_F = (V_F, F, c)$ be the subgraph of G induced by F, associated to the RPP problem. We define the TSP graph, $G_{TSP} = (V_P, F_P, c_P)$, associated to the RPP, as a complete graph where V_P is the set of middle points of the edges in F, and F_P includes the edges among the points in V_P. The cost associated to the edges is the Euclidean distance between their corresponding end points.

Although we transform a NP-hard problem in another one, we generate a TSP with less nodes than the RPP ($|V_p| = |F| = r$), which allows us to solve bigger RPP problems.

3.2 The SOFM Algorithm Applied

The network proposed to solve the TSP will have a competitive layer with a variable number of neurons to which neurons can be added or removed dynamically. This network is broadly discussed in [7]. Initially, the competitive layer will be defined by a vector of neurons of a size bigger than or equal to the number of cities in the problem. When a neuron wins for more than one city, a new neuron will be added next to it. To avoid that the number of neurons increases unlimitedly, the not winner neurons will be eliminated.

The steps of the proposed algorithm are shown below:

```
Repeat N_ITER times
 Repeat r times
    Select a city randomly as the next input to the network
    Determine the winner neuron, g
    Update the weights of g and its neighborhood
 end-repeat
 If each city has a neuron at a distance smaller than MAX_ERR
     END
 end-if
 Insert and remove the required neurons
 Update the learning rate and the neighborhood radius
end-repeat
Remove the crosses from the path
```

The algorithm stops if `N_ITER` loops have been completed or if there exist a neuron at a distance smaller than a prefixed error, `MAX_ERR`, for each city in the problem, what first occurs.

After the application of the algorithm, a second phase of fine-tuning is performed, with the same steps, but considering a learning rate and a radius of neighborhood with constant values, equals to the final values of the previous phase. They will be performed as many iterations as cities are in the problem, showing these by rotation. If after the fine-tuning phase the distance between the neurons and the associated cities is not smaller than the error, successive phases of fine-tuning can be applied.

Addition and Removal of Neurons. One of the inconveniences observed when applying the basic algorithm proposed by Kohonen is that it may exist neurons that does not win for any city, while others win for several. To solve this problem, in the self-organization process we will add or remove neurons when necessary. If the same neuron becomes the winner for several cities, it will be duplicated, so as in the next iteration of the algorithm both neurons can distribute for themselves the cities for which the initial neuron was the winner. If a neuron does not win for any city after various iterations of the algorithm, it is removed.

Let `ITER` be the current iteration of the training algorithm, `WIN(k)` the counter of victories of the neuron `k`, `LAST(k)` the last iteration for which the neuron `k`

won, `w(k)` the weights of the neuron `k` and `T` a random value between 0 and 1. The algorithm applied to add and remove neurons is the following:

```
For each neuron k of the output layer
   If WIN(k)>1
     If WIN(k-1) > WIN(k+1)
       Insert a neuron X between neurons k and k-1, with weights:
              w(X) = 0,04*w(k-1) + 0,95*w(k) + 0,01*T
     else
       Insert it between neurones k and k+1, with weights:
              w(X) = 0,04*w(k+1) + 0,95*w(k) + 0,01*T
     end-if
     Set LAST(X) = ITER, w(k) = 0,99*w(k)
   end-if
   If (WIN(k)=0) and (it was winner for no city when all the
      set of cities was presented) and (LAST(k) < ITER - 3)
          Remove neuron k
   end-if
end-repeat
```

The weights of the new neurons are calculated as described to allow they to be placed between the weights of their immediate neighbors. The weight of the multiple-winner has more importance since the cities for which it won are nearer to it than to the immediate neighbor considered. A random value is added, with a very small weight, to give the new neuron more freedom of adaptation. The weights of the multiple winner are slightly modified to give to the new neurons a bigger chance of victory in successive iterations of the training algorithm.

We must observe that before starting the phase of insertion/elimination of neurons the winner for each city has been determined (at the step of the algorithm that checks if for each city there is a neuron at a distance smaller than the defined error). This information is taken into account at the moment of removing a neuron. If cities are presented randomly, there is the possibility of no presentation of a city at a certain iteration, that implies that if there is a neuron near to this city, its last victory will be at most equal to the previous iteration and its counter of victories can be 0. It is also possible that this same city has not been presented during the last loops of the algorithm. Therefore, if we only consider the counters of the victories and the last victory of the neurons, this neuron could be removed, despite having an almost perfect adjustment to the not presented city. To prevent this situation, before removing the neuron we also check that it has not won when presenting all the set of patterns.

Crosses Removal. If in the graphical representation of a path for a TSP we observe the existence of crosses, this means that the solution found is not optimal. There will always be at least a shorter solution: the one that does not include any crosses. To eliminate crosses in the map, we apply 2-OPT exchange [17].

3.3 From the TSP Solution to the RPP Solution

Let be $\{p_1, p_2, ..., p_r\}$ the sequence of points of the TSP solution. For each TSP point, p_a, we know the end nodes of the RPP edge associated to it, (o_{p_a}, d_{p_a}).

We take p_1, p_2 and p_3 and we determine the way to connect edges (o_{p_1}, d_{p_1}), (o_{p_2}, d_{p_2}), (o_{p_3}, d_{p_3}) to generate a sequence of minimum cost. Then we add to the RPP path the end points of the first edge, in the order defined by the previous sequence. Next, we determine the best way to connect the RPP edges associated to p_2 and p_3 with the last edge included in the RPP path, and we add to the RPP path the end points of the edge associated with p_2. Like in the first case, we must connect 3 RPP edges, but now the order of the end points of the first one is fixed. The process follows in a similar way until the RPP edge associated to the point p_{r-1} has been included in the RPP path. Then we only have to insert the edge (o_{p_r}, d_{p_r}), connecting it with the first and the last points included in the RPP solution, in the order that generates a sub-path of minimum cost.

4 Computational Results

The proposed algorithm was coded in C and run on a Linux system with 1.5GHz Intel Centrino processor and 512M RAM.

We have generated a set of sample problems with a variable number of points and required edges. For each problem we made 10 proofs considering the following neighborhood functions: rectangular, pipe, triangular, Gaussian, Mexican-hat and exponential-concave. The Gaussian, pipe and triangular functions generate the shortest paths in a bigger number of problems. The best times are obtained with the rectangular function. In general, the triangular function generates the shortest solutions in less time. The computational results are summarized in Table 1. For each problem we specify the number of nodes, $|V|$, and required edges, $|F|$, of the RPP, the cost of the required edges, C_F, the best solution found by the proposed method, $Best_{NN}$, and the time in seconds to reach that solution, T_{NN}.

Obviously, when we add or remove neurons, we are modifying the structure of the map, and this has an effect on the natural self-organization process. That is, we must attempt that the number of insertions/removals be as small as possible. The non-winning neurons must be removed not too soon and not too late. If we eliminate a neuron that has not won when the current iteration comes to end, this can have happened because in that iteration was not presented the city for which it was the winner (if cities are presented randomly). It is possible that the adjustment between the neuron and the city were almost perfect but, if we remove it, in later iterations it will be necessary to add a new neuron which will try to adjust this city. If neurons are removed too late, we would dragging an unnecessary load which supposes a greater consumption of memory and a bigger processing time.

It has been proved that the map spreads out more quickly if the insertion and elimination are made, if it is necessary, at the end of each loop of the SOFM algorithm. If the problem is being solved correctly, the number of neurons

added/removed will decrease as the iterations proceed, until reaching a point in which the number of neurons is equal to the number of cities. Although the algorithm worked better before including the process of removing of neurons, it is necessary to limit the growth of the map and, with this, the increase in the space of memory and the calculation time needed.

Other proportions different from the used in Section 3.2 have been proved in the assignment of weights to the new neurons, but the best results have been reached with that combination $(4\%, 95\%, 1\%)$.

Table 1. Some results

Problem	$\|V\|$	$\|F\|$	C_F	$Best_{NN}$	T_{NN}	$Best_{Ants}$	T_{Ants}
p01	7	4	20.22	47.21	0	46.88	0
p02	7	4	36.99	56.06	0	56.06	0
p03	7	4	38.56	48.76	0	48.76	0
p04	7	4	17.67	45.30	0	45.30	1
p05	12	12	1200.00	1200.00	0	1200.00	2
p06	12	16	1600.00	1600.00	1	1600.00	2
p07	12	12	283.14	283.14	1	283.14	0
p08	24	12	120.00	240.00	0	240.00	0
p09	24	24	240.00	240.00	0	240.00	2
p10	24	28	320.00	320.00	0	320.00	1
p11	24	24	409.71	449.71	0	449.71	0
p12	92	92	116.28	116.28	9	116.28	18
p13	92	76	84.28	116.28	5	116.28	10
p14	92	92	118.63	132.28	10	132.28	16

To evaluate the proposed solution, we compare it with another one we developed that applies artificial ants to solve the same problem [18]. The ant-based algorithm was proved using a set of 26 benchmark instances proposed by Chistofides and Corberán, [9], [10], generating very good results. The last two columns in Table 1 show the best solution obtained by this method, $Best_{Ants}$, and the time associated, T_{Ants}, (running on the same computer).

We observe that the cost of the solution is similar for both algorithms, but the SOFM-based algorithm requires less time than the other.

5 Conclusions

We present an algorithm to solve the undirected Rural Postman Problem, based on the use of self-organizing feature maps. This neural network model can be applied to postman problems defined on the Euclidean plane. Up to the authors' knowledge, there is no paper dealing with the application of this neural network model to solve this problem.

The method always generates a feasible solution for the problem. Another advantage of this solution is that no mathematical formulation of the problem

is required to reach a solution; we only require the graph representation of the problem.

In the future, we will attempt to apply this method to bigger problems.

References

1. Orloff, C.S.: A Fundamental Problem in Vehicle Routing. Networks 4, 35–64 (1974)
2. Eiselt, H.A., Gendreau, M., Laporte, G.: Arc Routing Problems, Part II: The Rural Postman Problem. Operations Research 43, 399–414 (1995)
3. Grötschel, M., Jünger, M., Reinelt, G.: Optimal Control of Plotting and Drilling Machines: a Case Study. Mathematical Methods of Operations Research 35, 61–84 (1991)
4. Ghiani, G., Improta, G.: The Laser-Plotter Beam Routing Problem. Journal of the Operational Research Society 52, 945–951 (2001)
5. Lenstra, J.K., Rinnooy-Kan, A.H.G.: On General Routing Problems. Networks 6, 273–280 (1976)
6. Lawler, E.L., Lenstra, J.K., Rinnooy-Kan, A.H.G., Shmoys, D.B. (eds.): The Traveling Salesman Problem. John Wiley and Sons, Chichester (1985)
7. Pérez-Delgado, M.L.: Los Mapas de Rasgos Auto-Organizativos y el Problema del Viajante de Comercio. Ediciones Universidad de Salamanca, Spain (2004)
8. Pérez-Delgado, M.L.: Sistemas Multiagente para la Resolución del Problema del Viajante de Comercio. IEEE Latin America Transactions 2(1) (2004)
9. Christofides, N., Campos, V., Corberán, A., Mota, E.: An Algorithm for the Rural Postman Problem. Imperial College Report. IC.O.R.81.5, London (1981)
10. Corberán, A., Sanchis, J.M.: A Polyhedral Approach to the Rural Postman Problem. European Journal of Operational Research 79, 95–114 (1994)
11. Ghiani, G., Laporte, G.: A Branch and Cut Algorithm for the Undirected Rural Postman Problem. Mathematical Programming 87, 467–481 (2000)
12. Letchford, A.N.: Polyhedral Results for some Constrained Arc Routing Problems. PhD Dissertation, Lancaster University. Lancaster (1996)
13. Frederickson, G.: Approximation Algorithms for some Postman Problems. Journal of the ACM 26, 538–554 (1979)
14. Groves, G.W., van Vuuren, J.H.: Efficient Heuristics for the Rural Postman Problem. Orion 21(1), 33–51 (2005)
15. Baldoquín, M.G.: Heuristics and Metaheuristics Approaches Used to Solve the Rural Postman Problem: A Comparative Case Study. Dpto. Matemática General, Facultad Ing. Industrial, Universidad Técnica de La Habana, Cuba
16. Kohonen, T.: Self-Organization and Associative Memory. Springer Series in Information Sciences, vol. 8. Springer, Heidelberg (1982)
17. Gendreau, M., Hertz, A., Laporte, G.: New Insertion and Postoptimization Procedures for the Traveling Salesman Problem. Operations Research 40, 1086–1094 (1992)
18. Pérez-Delgado, M.L., Calvo, J., Matos-Franco, J.C., Pérez-Iglesias, J.L.: The Undirected Rural Postman Problem Solved by Artificial Ants. In: Proc. 11th International Conference on Computer Aided Systems Theory, Las Palmas de Gran Canaria. Spain, pp. 279–280 (2007)

Optimization Methods for Large-Scale Production Scheduling Problems

Roland Braune, Stefan Wagner, and Michael Affenzeller

Upper Austria University of Applied Sciences
Hauptstrasse 117
4232 Hagenberg
{roland,stefan,michael}@heuristiclab.com

Abstract. In this paper we present a computational study of optimization methods for production scheduling problems which can be described by a job shop model. Contrary to most existing publications in this field our research focuses on the performance of these methods with respect to large-scale problem instances. The examined methods rely on a graph model as a solution representation and have originally been designed for problems of small size. We apply them to a set of semi-randomly generated problem instances whose properties have been transferred from common (smaller) benchmarks. The experiments are based on tardiness minimization and the results are evaluated in relation to a priority rule based heuristic.

1 Introduction

A common way of describing production scheduling problems in an abstract form is to reduce them to a job shop model [5]. This model is widely discussed in literature and has been subject to intensive methodic research over the past decades. In a job shop production jobs are subdivided into operations. Each such operation is assigned a particular machine on the shop floor on which it has to be processed. The order in which a job's operations are executed is predetermined and can be different for each job. This is also referred to as the *technological order* of operations or as the *precedence constraints* of the associated scheduling problem. Each machine can process at most one operation at a time (*capacity constraint*) and there is no preemption allowed which means that once the execution of an operation has started, it cannot be interrupted until it is finished.

We denote the set of all jobs by $J = \{j \mid 1 \leq j \leq n\}$ and the set of all machines by $M = \{k \mid 1 \leq k \leq m\}$. It is assumed that each job has to be processed on every machine, hence $O = \{o_{jk} \mid 1 \leq j \leq n, 1 \leq k \leq m\}$ constitutes the set of all operations. Operation o_{jk} refers to the operation of job j which is to be executed on machine M_k. Each operation o_{jk} requires p_{jk} time units to execute, which we call the operation's processing time. A *feasible schedule* S is a set of starting times s_{jk} for each operation which satisfies both, the precedence and the capacity constraints. The problem of finding a schedule which is as good as

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 812–819, 2007.
© Springer-Verlag Berlin Heidelberg 2007

possible with regard to a given objective function is considered a combinatorial optimization problem.

Given a particular schedule S, the completion time of each job j is denoted by C_j. The minimization of the *makespan* $C_{max} = \max(C_1, \ldots, C_n)$ is the most common objective in the area of job shop scheduling. However, in our contribution we use a different objective function, namely the total weighted tardiness. In this context, each job is assigned a due date d_j and a weight w_j. The tardiness T_j of a job j is defined as $T = \max(0, C_j - d_j)$. The optimization problem consists in minimizing the expression $\sum_{j=1}^{n} w_j T_j$. In the 3-field notation of Graham et al. [7] this problem is denoted by $Jm||\sum_{j=1}^{n} w_j T_j$.

The considered problem is $\mathcal{NP}$-hard since it is a generalization of the single-machine problem $1||\sum_{j=1}^{n} w_j T_j$ which is known to be strongly $\mathcal{NP}$-hard [9].

Literature on $Jm||\sum_{j=1}^{n} w_j T_j$ is quite sparse. Priority rule based approaches are described in [15] and [2]. Singer and Pinedo propose a Branch&Bound method [14] and a shifting bottleneck procedure [12]. In [8] Kreipl describes an Iterated Local Search method for tardiness job shops. Furthermore, Genetic Algorithms (e.g. [11]) and Tabu Search (e.g. [3]) have also been applied to this kind of problem.

In this contribution, we focus on a comparison of selected methods for tardiness minimization in job shops. Since most of the research in this field concentrates on relatively small problems (up to 400 operations), we investigate how these methods perform when applied to much larger instances (up to 4000 operations) as they display the situation of a real-world scenario much closer.

In Section 2 we introduce the graph model for job shop problems. The methods which are subject to our investigations are outlined in Section 3. Section 4 describes in detail how the benchmark problem set was generated followed by the discussion of the experimental results (cf. Section 5). Conclusions and outlook on further research are given in the final Section 6.

2 Graph Model

A job shop problem can be mapped to a disjunctive graph $G = (N, C, D)$ [13]. N denotes the node set, which contains one node for each operation o_{jk}, one source node U and n sink nodes $V_1, \ldots, V_n$ (one for each job). C is the set of directed (conjunctive) arcs and represents the technological order of operations whithin each job. The disjunctive arc set D consists of pairs of directed arcs between each two operations on the same machine. Each node is assigned a weight which corresponds to the operation's processing time. Figure 1 shows a disjunctive graph for a problem involving 3 jobs and 3 machines (3×3).

A schedule can be obtained by selecting one of two possible arcs between each pair of operations on the same machine. The result, which is also referred to as a (complete) *selection*, represents a directed graph. The schedule is feasible if the corresponding graph is acyclic.

Given a complete selection, one can compute the longest path $L(U, V_j)$ from the source to sink node V_j. The length of the longest path equals the completion

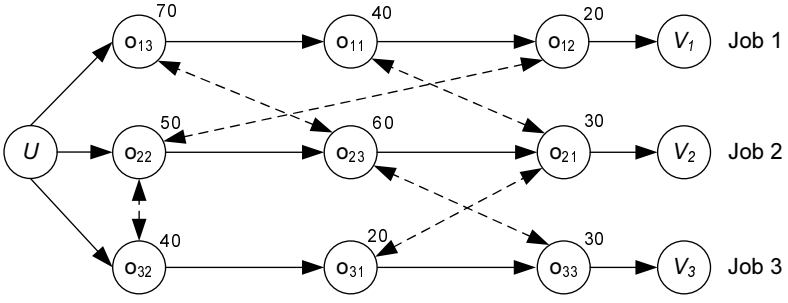


Fig. 1. Disjunctive Graph for a 3×3 Problem

time C_j of job j. A path $L(U, V_j)$ is called *critical* if it leads to a completion time $C_j > d_j$ and thus to job tardiness.

3 Methodology

Exact solution methods in the area of job shop scheduling are limited to small problems up to 400 operations. Heuristic approaches on the other hand have the advantage of being applicable to much larger instances. In our contribution, we therefore selected two techniques which appeared quite promising for such problem sizes: The Shifting Bottleneck Procedure by Pinedo and Singer [12] in a slightly modified form and the Iterated Local Search algorithm by Kreipl [8]. Both methods are based on the graph model described in Section 2 and their most important characteristics are outlined in the following two sections.

3.1 The (Modified) Shifting Bottleneck Procedure

The Shifting Bottleneck Procedure (SBP) is a problem decomposition approach originally proposed for the minimum makespan job shop [1]. It splits the original problem into m single machine problems and sequences them (separately) one after the other in bottleneck order. An outline of the basic principle is given in Algorithm 1, where $M_0 \subseteq M$ denotes the set of already scheduled machines.

We basically implemented the Shifting Bottleneck Procedure for tardiness scheduling as described in [12] except for the subproblem optimization. Pinedo and Singer propose a partial enumeration heuristic for the solution of the single machine problems. However, due to excessive running times this method is impractical for large single machine instances involving 50 or more operations. Especially when using enhanced control structures such as reoptimization or backtracking the efficient optimization of subproblems is of great importance. Even metaheuristics such as Tabu Search or Genetic Algorithms turned out to be too time consuming in this special context. Therefore we applied a simple first improvement local search algorithm based on principles described in [4] with modifications concerning the handling of delayed precedence constraints and the specific objective function.

Algorithm 1. Shifting Bottleneck Procedure - Main Flow

```
Initialize disjunctive graph G
Set M_0 ← ∅
while M_0 ≠ M do
  for all k ∈ M \ M_0 do
    Formulate single machine (sub)problem
    Solve single machine problem
  end for
  Select bottleneck machine k' ∈ M \ M_0
  Insert directed arcs for machine k' into G
  Set M_0 ← M_0 ∪ {k'}
  Reoptimize scheduled machines (optional)
end while
```

3.2 The Iterated Local Search Algorithm

Iterated Local Search (ILS) or Large Step Optimization methods in the job shop scheduling domain have been extensively studied first in [10]. The main idea behind this kind of local search algorithms is the alternation of intensification and diversification phases. In the intensification or small step phase usually a simple neighborhood search (descent) algorithm gets applied in order to find the nearest local minimum. The diversification or large step phase on the other hand is responsible for exploring the solution space in order to find new promising regions.

For our experiments we implemented the ILS algorithm for the total weighted tardiness job shop problem proposed by Kreipl [8]. The method uses a neighborhood structure which is based on the graph model as described in Section 2. Neighboring solutions are basically obtained by interchanging operations which are on at least one critical path in the directed graph corresponding to the current solution. In order to compact the schedule after the interchange, two further related operations may be interchanged.

In order to escape from local minima, the large step phase needs to accept neighboring solutions which are worse than the current solution. This process is controlled by a Metropolis algorithm with fixed temperature. The number of iterations is determined depending on how close the previous small step phase got to the best solution reached so far.

4 Problem Setup

Benchmark problems for tardiness job shops are rare. For this reason, it is of common practice to modify well-known makespan related benchmark problems by adding due dates and weights to them. Pinedo and Singer [12] used instances from the OR-Library[1] and adapted them in the following way: Given a *due date tightness factor* f, the due dates d_j are computed as $d_j = r_j + \lfloor f \cdot \sum_{k=1}^{m} p_{jk} \rfloor$,

[1] `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`

where $r_j = 0$ is assumed for all jobs. For 20% of all jobs they choose $w_j = 4$, 60% are assigned $w_j = 2$ and the remaining 20% are set to $w_j = 1$.

Since Pinedo and Singer used only 10×10 instances, and the OR-library problems are generally of limited size, we had to create our own benchmarks. Additionally, our aim was to focus on congested job shops involving much more jobs than machines (rectangular job shops). For our experiments we applied four different problem configurations: $\{50, 100\}$ jobs on 10 machines and $\{100, 200\}$ jobs on 20 machines.

Similar to most of the OR-Library instances, the operation processing times were sampled from a uniform distribution. However, as for the machine sequences (technological orders) a different approach has been chosen: The machine sequences, often also referred to as *job routings*, were created according to predefined template distributions. In real world manufacturing scenarios, job routings are also not completely different from each other: Some machines are always passed through in the same order, other machines are dedicated entry or exit points, etc. Although most of the OR-Library instances have job routings sampled from a uniform distribution, the resulting actual machine positions within the routings are not evenly distributed due to the small number of samples. What looks like a drawback at first sight, is exactly what we want: A certain degree of similarity between job routings.

For our experiments we extracted positional information from the original OR-Library instances and transferred it to our newly generated large-scale instances. This course of action has the following advantage: The resulting problems are "similar" to the original problems. The transfer of machine positions preserves some important properties, especially with regard to problem difficulty in the context of the Shifting Bottleneck Procedure, which permits a better interpretation of results.

5 Experimental Results

The computational experiments were carried out using randomly generated problems of size 50×10, 100×10, 100×20 and 200×20. The job routings are based on 5 different benchmark instances taken from the OR-library according to Section 4: `abz5`, `ft10`, `la19`, `orb03` and `orb10`. The experiments were coded in C# in the HeuristicLab optimization environment [16] and run on a 3 GHz Pentium 4 PC with 1 GB RAM.

The parameters for the Shifting Bottleneck Procedure are summarized in Table 1. It has to be remarked that the SB procedure was not able to produce reasonable results for instances greater than 100×10. Furthermore, the backtracking control structure could only be applied to the 50×10 instance. In the 100-job case the computational effort needed to solve the subproblem instance was too high to allow backtracking in reasonable time. Therefore the 100-job instances were run with reoptimization only.

The Iterated Local Search algorithm has been applied to all instances under the same parameter settings. Modifications concerning the number of iterations

Table 1. Parameter Settings for the Shifting Bottleneck Procedure

	50×10	100×10
Subproblem solution method	FI Descent	FI Descent
Subproblem ATC parameter	2	4
Reoptimization	yes	yes (2x)
Backtracking	yes	no
Backtracking aperture size	2	-

Table 2. Due Date Tightness Factors

Size	f_{tight}	f_{loose}
50×10	4	4.5
100×10	7	8
100×20	4.5	5
200×20	8	9

Table 3. Algorithm Running Times

Size	**SB (avg.)**	**ILS (fixed)**
50×10	338 sec.	300 sec.
100×10	90 sec.	600 sec.
100×20	-	1200 sec.
200×20	-	2400 sec.

in the diversification phase did not yield consistently better results, not even for the large instances. Neither did the extension of time limits.

Besides the methods described in Section 3, we applied a priority rule based heuristic as a comparison baseline. The heuristic is based on a *non-delay* scheduling algorithm [6] and it uses the *Apparent Tardiness Cost* (ATC) rule [15] for conflict set resolution.

Table 4 shows the computational results for our benchmark set. The two optimization algorithms have been applied to the problems under tight and loose due dates respectively (cf. Table 2). The due date tightness factors f have been determined experimentally such as to fit for all five instances.

Concerning the running times (cf. Table 3) we first measured the average running time for the SB procedure applied to the 50×10 instances. We then used this value as a base time limit for the ILS algorithm. Anyway, the SB running time for the 100×10 problems is significantly lower. This is due to the absence of the backtracking control structure as explained above.

It can be observed that the Shifting Bottleneck Procedure does not show a constant performance for the tackled problems. Especially the results for the loose due date problems are not consistent, since in many cases they were worse than the ones obtained for tight due dates. As for the 100×10 instances, which were run without backtracking, no improvements at all could be gained over the rule based solution.

The ILS algorithm on the other hand achieves a considerably good solution quality, even for the tight due date case. For some instances the tardiness could even be minimized to zero, which corresponds to a 100 % improvement. However, as for the 200×20 instances with tight due dates, the algorithm seems to reach its limits. Improvements are constantly below 10 % and could not be significantly increased by parameter tuning nor by providing more computation time. We conjecture that only drastic changes to the algorithm such as different neighborhood structures may improve the performance for this kind of problems.

Table 4. Computational Results - Relative Percentage Deviation from Baseline

		Tight Due Dates		Loose Due Dates	
Base problem	**Size**	**SB**	**ILS**	**SB**	**ILS**
abz5	50 × 10	21.91 %	-26.69 %	-14.06 %	-55.82 %
	100 × 10	51.11 %	-21.02 %	78.35 %	-68.66 %
	100 × 20	-	-27,39 %	-	-100,00 %
	200 × 20	-	-6,54 %	-	-36,47 %
la19	50 × 10	33.14 %	-28.69 %	62.26 %	-53.71 %
	100 × 10	65.09 %	-13.19 %	85.98 %	-25.08 %
	100 × 20	-	-18,99 %	-	-100,00 %
	200 × 20	-	-4,77 %	-	-62,86 %
ft10	50 × 10	-1.31 %	-39.02 %	-24.95 %	-55.22 %
	100 × 10	70.65 %	-20.99 %	91.36 %	-41.25 %
	100 × 20	-	-22,44 %	-	-34,05 %
	200 × 20	-	-5,08 %	-	-8,42 %
orb10	50 × 10	1.49 %	-37.57 %	-29.67 %	-60.35 %
	100 × 10	42.96 %	-21.31 %	118.56 %	-52.47 %
	100 × 20	-	-22,51 %	-	-63,20 %
	200 × 20	-	-6,01 %	-	-22,14 %
orb03	50 × 10	54.49 %	-21.63 %	63.54 %	-44.02 %
	100 × 10	214.28 %	-18.19 %	396.09 %	-27.62 %
	100 × 20	-	-12,03 %	-	-29,60 %
	200 × 20	-	-3,77 %	-	-8,95 %

6 Conclusions and Perspective

We have carried out a performance analysis of two different optimization methods for large-scale tardiness job shops. On the basis of benchmark problem instances which have been created in a semi-random manner for this purpose, we can draw the following main conclusions:

- The modified Shifting Bottleneck Procedure shows unstable and inconsistent behaviour with respect to solution quality.
- The performance of the SB procedure seems to strongly depend on backtracking.
- The SB procedure works well for small problems. However, the method turned out to be impractical for problems greater than 50 × 10, at least for the problem structure and objective function used in this paper.
- The ILS algorithm obtained significant improvements for problems up to 2000 operations - without any modifications to the basic version.
- Problems involving more than 4000 operations are considerably harder to solve for the ILS method. The neighborhood structure we used seems to be insufficient for such a huge solution space.

Our future research in this area will strongly focus on Iterated Local Search methods and their further enhancement for large-scale problem instances. In

essence this will include the development and analysis of new neighborhood concepts for tardiness job shops and their effective embedding into the algorithms.

References

1. Adams, J., Balas, E., Zawack, D.: The shifting bottleneck procedure for job shop scheduling. Management Science 34(3), 391–401 (1988)
2. Anderson, E.J., Nyirenda, J.C.: Two new rules to minimize tardiness in a job shop. International Journal of Production Research 28(12), 2277–2292 (1990)
3. Armentano, V.A., Scrich, C.R.: Tabu search for minimizing total tardiness in a job shop. International Journal of Production Economics 63(2), 131–140 (2000)
4. Braune, R., Affenzeller, M., Wagner, S.: Efficient heuristic optimization in single machine scheduling. In: Bruzzone, A., Guasch, A., Piera, M., Rozenblit, J. (eds.) Proceedings of the International Mediterranean Modelling Multiconference I3M 2006, Piera, LogiSim, Barcelona, Spain, pp. 499–504 (2006)
5. French, S.: Sequencing and Scheduling: An Introduction to the Mathematics of the Job Shop, Chichester, Horwood et al. (1982)
6. Giffler, B., Thompson, G.L.: Algorithms for solving production scheduling problems. Operations Research 8, 487–503 (1960)
7. Graham, R.L., Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G.: Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Operations Research 5, 187–326 (1979)
8. Kreipl, S.: A large step random walk for minimizing total weighted tardiness in a job shop. Journal of Scheduling 3, 125–138 (2000)
9. Lenstra, J.K., Rinnooy Kan, A.H.G., Brucker, P.: Complexity of machine scheduling problems. Annals of Discrete Mathematics 1, 343–362 (1977)
10. Lourenco, H.R.: Job shop scheduling: Computational study of local search and large step optimization methods. European Journal of Operational Research 83, 347–364 (1995)
11. Mattfeld, D.C., Bierwirth, C.: An efficient genetic algorithm for job shop scheduling with tardiness objectives. European Journal of Operational Research 155(3), 616–630 (2004)
12. Pinedo, M., Singer, M.: A shifting bottleneck heuristic for minimizing the total weighted tardiness in a job shop. Naval Research Logistics 46, 1–17 (1999)
13. Roy, B., Sussmann, B.: Les problèmes d'ordonnancement avec contraintes disjonctives. Note D.S. no. 9 bis, SEMA, Paris, France, Décembre (1964)
14. Singer, M., Pinedo, M.: A computational study of branch and bound techniques for minimizing the total weighted tardiness in job shops. IIE Transactions 30(2), 109–118 (1997)
15. Vepsalainen, A.P.J., Morton, T.E.: Priority rules for job shops with weighted tardiness costs. Management Science 33(8), 1035–1047 (1987)
16. Wagner, S., Affenzeller, M.: Heuristiclab: A generic and extensible optimization environment. In: Proceedings of the 7th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA '05). Springer Computer Science, pp. 538–541. Springer, Heidelberg (2005)

Self-adaptive Population Size Adjustment for Genetic Algorithms

Michael Affenzeller[1], Stefan Wagner[1], and Stephan Winkler[2]

[1] Upper Austrian University of Applied Sciences
Department of Software Engineering
[2] Research Center Hagenberg
Softwarepark 11, 4232 Hagenberg, Austria
{michael,stefan,stephan}@heuristiclab.com

Abstract. Variable population sizing techniques are rarely considered in the theory of Genetic Algorithms. This paper discusses a new variant of adaptive population sizing for this class of Evolutionary Algorithms. The basic idea is to adapt the actual population size depending on the actual ease or difficulty of the algorithm in its ultimate goal to generate new child chromosomes that outperform their parents.

1 Introduction

The effects of population size of Evolutionary Algorithms (EAs) has been subject of intensive research by the EA community since its beginning. Already in the first proposed variants of Evolution Strategies (ES), the community paid a lot of attention to the appropriate choice for μ (population size) and λ (birth surplus) for the certain variants of (μ, λ)-ES and $(\mu + \lambda)$-ES, respectively (for an overview see [5], e.g.). In the Genetic Algorithm (GA) community, too, well-known researchers concentrated on the effects of population size with respect to achievable solution quality right from the beginning [8]. All these considerations assume the population size to be constant during the run of an EA. However, there are many hints in population genetics and also in nature itself that underpin the reasonability of a variable population size during the evolution of a certain species or during the run of an EA, respectively.

The Genetic Algorithm with Variable Population Size (GAVaPS) [3] eliminates the population size as an explicit parameter by introducing the properties "age" and "maximal lifetime" [3]. A further strategy to introduce adaptive population sizing schemes is given by the so called APGA (Adaptive Population size GA) [4] which is based upon a steady state GA and adopts the lifetime of an individual as it stands. Eiben et al. [6] introduce a growing population size in case of high fitness improving capacity or in case of longer lasting stagnation. Short stagnation periods cause the decrease of population size.

The adaptive population sizing scheme presented in this article follows a basically different approach by interpreting the actual population size as a consequence of the capacities of the actually applied reproduction operators (crossover

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 820–828, 2007.
© Springer-Verlag Berlin Heidelberg 2007

and mutation) to create offspring that outperform the solution quality of their own parents. As long as it is possible to generate new and in this sense successful individuals, the next generation grows. Therefore, apart from an upper limit that comes with the limited environmental resources, the respective next generation's population is filled up with new individuals as long as the allele pool of the present generation is not yet fully utilized.

This article is organized as follows: Section 2 describes premature convergence on the basis of a typical GA application and aims to the bottom of reasons for premature convergence. The goal of Section 3 is to use these findings in order to discuss generic algorithmic extensions and also to describe their concrete implementation. The ability of these further developed algorithmic concepts to preserve essential genetic information more efficiently is demonstrated on the basis of a concrete example that has already been used in Section 2. Finally, Section 4 sums up the main results of the paper and states some perspectives for ongoing future research.

2 Premature Convergence and Its Reasons

GAs are also frequently faced with a problem which, at least in its impact, is quite similar to the problem of stagnating in a local but not global optimum. This drawback, called premature convergence in the terminology of GAs, occurs if the population of a GA reaches such a suboptimal state that the genetic solution manipulation operators (crossover and mutation) are no longer able to produce offspring that are able to outperform their parents (e.g. [7], [1]). In general this happens mainly when the genetic information stored in the individuals of a population does not contain that genetic information which would be necessary to further improve the solution quality. Therefore, in contrast to the present contribution, the topic of premature convergence is considered to be closely related to the loss of genetic variation in the entire population in GA-research [10], [11]. Here we do not identify the reasons for premature convergence in the loss of genetic variation in general but more specifically in the loss of what we call essential genetic information, i.e. in the loss of alleles which are part of a global optimal solution.

This basic concept of a GA poses several questions and associated problems:

- Is crossover always able to fulfil the implicit assumption that two above average parents can produce even better children?
- Which of the available crossover operators is best suited for a certain problem in a certain representation?
- Which of the resulting children are "good" recombinations of their parents chromosomes?
- What makes a child a "good" recombination?
- Which parts of the chromosome of above average parents are really worth being preserved?

In order to get to the bottom of the question how a Genetic Algorithm could optimally utilize the available gene pool, let us first consider the internal processes of a very typical simple GA applied to a typical combinatorial optimization problem, namely a 130 city benchmark Traveling Salesman Problem (TSP). By reasons of compactness, the results are mainly shown on the basis of diagrams and give only a brief description of introduced operators, parameter settings, and test environments. Furthermore, the chosen benchmark instance (the ch130 benchmark TSP) is of rather small dimension in order to allow the observation of essential alleles during the run of the algorithm.

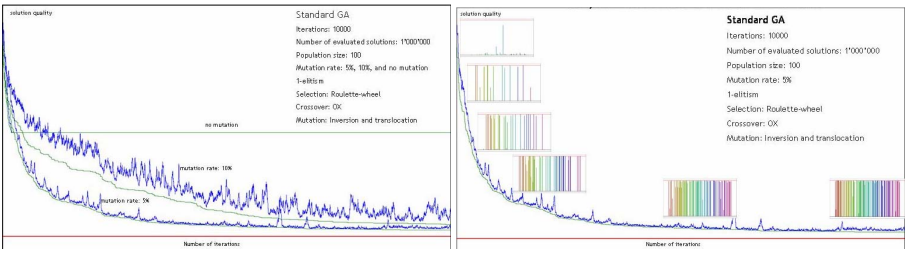


Fig. 1. The effect of mutation for certain mutation rates (left diagram) and the distribution of essential genetic information for a mutation rate of 5% (right diagram) both in case of a standard GA for the ch130 benchmark TSP

The results displayed in Fig. 1 (left diagram) show the effect of mutation for reintroducing already lost genetic information. The horizontal line of the diagram shows the number of iterations and the vertical line stands for the solution quality. The bottom line indicates the global optimal solution which is known for this benchmark test case. The three curves of the diagram show the performance of a Genetic algorithm with no mutation, with a typical value of 5% mutation as well as a rather high mutation rate of 10%. For each of the three curves the lower line stands for the best solution of the actual population and the upper line shows the average fitness value of the population members. The results with no mutation are extremely weak and the quality curve stagnates very soon and far away from the global optimum; the best and average solution quality are the same and no further evolutionary process is possible - premature convergence has occurred.

The right diagram of Fig. 1 shows the distribution of essential alleles over the iterations for a standard GA with a mutation rate of 5%. The interesting thing is that some minor ratio of essential alleles (visualized by inserting bar charts which have to be interpreted as snapshots after a certain number of iterations approximately corresponding to the position in the figure) is rapidly fixed in the population and the majority of essential alleles which are still missing have disappeared in the entire population. During the further run of the algorithm it is only mutation which can reintroduce this essential genetic information. As is shown in Fig. 1, without mutation premature convergence occurres at this early

state of evolutionary search. But with an appropriate mutation rate (5% in this example) more and more essential alleles are discovered ending up with quite a good solution.

These observations show that the role of mutation may be much more important than usually considered in GA literature where mutation is often described as a background operator. As our small example has shown, mutation can also hide severe weaknesses of certain crossover operators w.r.t. their inability to preserve the essential genetic information during the run of a GA.

In the following section a new generic GA variant is introduced that uses adaptive population sizing in combination with Offspring Selection [2] in order to bring out the best of the gene pool which is available at a certain generation of a GA.

3 Efficient Preservation of Essential Alleles by Adaptive Population Sizing

Assuming generational replacement as the underlying replacement strategy, the most essential question at generation i is, which parts of genetic information from generation i should be maintained in generation $i + 1$ and how this could be done most effectively applying the available information (chromosomes and according fitness values) and the available genetic operators selection, crossover and mutation.

The further developed algorithmic concepts based upon GA-solution manipulation operators aim to achieve this goal by trying to bring out as much progress from the actual generation as possible and loosing as little genetic diversity as possible at the same time.

The implementation of this idea is done with on the fly population size adjustment in that sense that potential offspring generated by the basic genetic operators are accepted as members or the next generation if and only if they are able to outperform the fitness of their own parents and if they are new in that sense that their chromosome consists of a concrete allele alignment that is not represented yet in an individual of the next generation. As long as new and w.r.t. the definition above "successful" individuals can be created from the gene pool of the actual generation, the population size is allowed to grow. A potential offspring which is not able to fulfill these requirements is simply not considered for the gene pool of the next generation

The right part of Fig. 2 represents the gene pool of the alleles at a certain generation, say i whereas the left part aims to illustrate how this genetic can be used in order to generate a next population i of a certain size which may be smaller or larger than the actual population i depending on how successful the genetic manipulation operators crossover and mutation are in their above stated stated claim to produce new and successful chromosomes.

For a generic, stable and robust realization of the ideas stated above, some practical aspects have to be considered and implemented additionally:



Fig. 2. The left part of the figure represents the gene pool at generation i and the right part indicates the possible sice of generation i+1 which must not go below a minimum size and also not exceed an upper limit which are user defined parameters

- The algorithm should offer different settings also for conventional parent selection. So the selection mechanisms for the two respective parents do not necessarily have to be the same. In many examples a combination of proportional (roulette-wheel) selection and random selection has already shown a lot of achievement potential (for example in combination with Genetic Programming [12]). It is also possible and reasonable with the algorithmic concepts described here to disable parent selection totally as scalable selection pressure comes along with the selection mechanisms after reproduction. This can be achieved by setting both parent selection operators to *Random*.
- The fact that reproduction results are only considered in case they are successful recombinations and eventually mutations of their parents chromosomes to use more than one crossover operator and more than one mutation operator at the same time. The reason for this possibility is given by the fact that only successful offspring chromosomes are considered for the ongoing evolutionary process which allows the application of crossover and mutation operators which do not produce good results mostly as long as they are still able to generate good offspring at least sometimes. On the one hand the insertion of such operators just increases the average selection pressure and therefore also the average running time, but on the other hand these operators can help a lot to broaden evolutionary search and therefore retard premature convergence. In case more than one crossover and mutation operator is allowed, the choice occurs by pure chance which has proven to produce better results than a preference of more successful operators.
- As indicated in Fig. 2, a lower as well as an upper limit of population size are still necessary in order to achieve efficient algorithmic performance. In case of a missing upper limit the population size would snowball especially in the first rounds which is inefficient. A lower limit of at least 2 individuals is also necessary as this indicates that it is no more possible to produce a sufficient amount of chromosomes that are able to outperform their own parents and therefore acts as a good detector for convergence.

- Depending on the problem at hand there may be several possibilities to fill up the next population with hopefully new individuals. If the problem representation allows an efficient check for structural identity it is recommendable to do this and accept new chromosomes as members for the next generation if there is no structurally identical individual yet. If a check for structural identity is not possible or too time-consuming there is still the possibility to assume two individuals as identical if they have the same fitness values as an approximative identity check.
- In order to terminate the run of a certain generation in case it is not possible to fill up the maximally allowed population size with new successful individuals, an upper limit of effort in terms of generated individuals is necessary. This maximum effort per generation is the maximum number of newly generated chromosomes per generation (no matter if it has been accepted or not).
- A further question is: When is a child chromosome better than its own parents? Is a child better, if it is better than the better of the two parents or is the child already better, if its better than the worse of the two parents? In order to answer this question, we have borrowed an aspect from Simulated Annealing: The Threshold fitness value that has to be outperform lies between the worse and the better parent and the user is able to adjust a lower starting value and a higher end value (*Comparison Factor Bounds*) where a *Comparison Factor Bound* value of 0.0 means that we consider the fitness of the worse parent and a *Comparison Factor Bound* of 1.0 means that we consider the better of the two parents. During the run of the algorithm, the *Comparison Factor Bound* is linearly scaled between the lower and the upper bound resulting in a broader search at the beginning and ending up with a more and more directed search at the end which picks up a basic idea of Simulated Annealing.

In order to inspect the consequences of the newly introduced algorithmic measures let us consider its effects when applying it to the 130-city TSP *ch130* and observe the allele frequencies over time as done for a standard GA in Section 2.

Fig. 3 shows the quality curve and the distribution of essential alleles for the new GA. When applying this GA with the new adaptive population sizing to the same benchmark test case as already treated in section 2, one can see that the global optimal solution is detected in only about 100 iterations. Nevertheless, the computational effort is comparable to the standard GA as much more individuals have to be evaluated at each iteration step due to the higher effort. Considering the distribution of essential alleles we see a totally different situation than in the case of the standard GA. Almost no essential alleles are lost and the ratio of essential alleles continuously increases in order to end up with a final population that contains almost all pieces of essential genetic information and therefore includes a very good solution. This shows that the essential alleles are preserved much more effectively and indicates that the influence of mutation is much smaller.

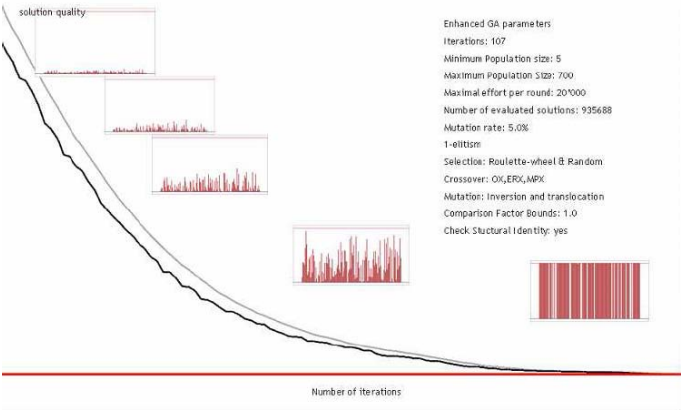


Fig. 3. The distribution of essential genetic information when using the enhanced GA with variable population sizing considering the ch130 benchmark TSP

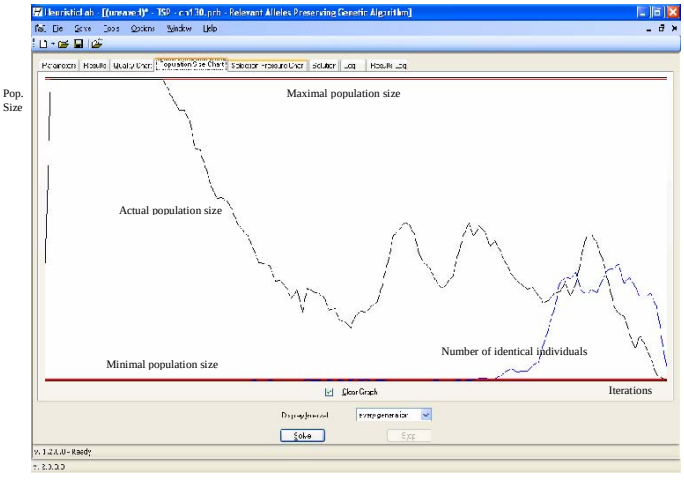


Fig. 4. Typical development of actual population size between the two borders (lower and upper limit of population size) displaying also the identical chromosomes that occur especially in the last iterations

4 Conclusion and Future Perspectives

The main aim of the present contribution is to give an idea of the potential of the presented adaptive population sizing mechanism. More sophisticated test series have to be performed and documented on the basis of well-known benchmark problems of different areas representing diverse fitness landscapes in order to demonstrate the capabilities and robustness of the algorithm.

The general strategy described in this paper guarantees that evolution is preserved mainly with crossover results that were able to mix the properties of their parents in an advantageous way in a sense that **survival of the fittest alleles is rather supported than survival of the fittest chromosomes** which is a very essential aspect for the preservation of essential genetic information stored in many individuals (which may not be the fittest in the sense of individual fitness). By this strategy, the essential genetic information stored in the population is preserved effectively during the run of the algorithm.

Some aspects of adaptive population sizing as presented in this paper are quite similar to the concept of Offspring Selection [2] and therefore it might be a fruitful idea to think about a parallel implementation of RAP-GA in a similar sense as SASEGASA [1] which is the parallel implementation based upon Offspring Selection.

References

1. Affenzeller, M., Wagner, S.: SASEGASA: A New Generic Parallel Evolutionary Algorithm for Achieving Highest Quality Results. Journal of Heuristics, Special Issue on New Advances on Parallel Meta-Heuristics for Complex Problems 10(3), 239–263 (2004)
2. Affenzeller, M., Wagner, S.: Offspring Selection: A New Self-Adaptive Selection Scheme for Genetic Algorithms. Adaptive and Natural Computing Algorithms, 218–221 (2005)
3. Arabas, J., Michalewicz, Z., Mulawka, J.: GAVaPS – A Genetic Algorithm with Varying Population Size. In: Proceedings of the First IEEE Conference on Evolutionary Computation, pp. 73–78. IEEE Computer Society Press, Los Alamitos (1994)
4. Baeck, T., Eiben, A.E., van der Vaart, N.A.L.: An Empirical Study on GAs Without Parameters. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) Parallel Problem Solving from Nature-PPSN VI. LNCS, vol. 1917, pp. 315–324. Springer, Heidelberg (2000)
5. Beyer, H.G.: The Theory of Evolution Strategies. Springer, Heidelberg (2001)
6. Eiben, A.E., Marchiori, E., Valk, V.A.: Evolutionary Algorithms with On-the-Fly Population Size Adjustment. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature - PPSN VIII. LNCS, vol. 3242, pp. 41–50. Springer, Heidelberg (2004)
7. Fogel, D.: An Introduction to Simulated Evolutionary Optimization. IEEE Trans. on Neural Networks 5(1), 3–14 (1994)
8. Goldberg, D.E.: Sizing Populations for Serial and Parallel Genetic Algorithms. In: Proceedings of the 3rd International Conference on Genetic Algorithms, pp. 70–79 (1989)
9. Reinelt, G.: TSPLIB - A Traveling Salesman Problem Library. ORSA Journal on Computing 3, 376–384 (1991)
10. Smith, R., Forrest, S., Perelson, A.: Population Diversity in an Immune System Model: Implications for Genetic Search. Foundations of Genetic Algorithms 2, 153–166 (1993)

11. Yoshida, Y., Adachi, N.: A Diploid Genetic Algorithm for Preserving Population Diversity - Pseudo-Meiosis GA. In: Davidor, Y., Männer, R., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature - PPSN III. LNCS, vol. 866, pp. 36–45. Springer, Heidelberg (1994)
12. Winkler, S., Affenzeller, M., Wagner, S.: Advanced Genetic Programming Based Machine Learning. Journal of Mathematical Modelling and Algorithms (2007)

Parallel Tabu Search and the Multiobjective Capacitated Vehicle Routing Problem with Soft Time Windows

Andreas Beham

Institute for Formal Models and Verification
Johannes Kepler University
Altenberger Str. 69
4040 Linz,Austria
abeham@heuristiclab.com

Abstract. In this paper the author presents three approaches to parallel Tabu Search, applied to several instances of the Capacitated Vehicle Routing Problem with soft Time Windows (CVRPsTW). The Tabu Search algorithms are of two kinds: Two of them are parallel with respect to functional decomposition and one approach is a collaborative multisearch TS. The implementation builds upon a framework called Distributed metaheuristics or DEME for short. Tests were performed on an SGI Origin 3800 supercomputer at the Johannes Kepler University of Linz, Austria.

1 Introduction

The VRP is a generalized name for a class of problems first formulated in the late 50s by Dantzig and Ramser [1]. The basic layout of the problem consists of a depot with several vehicles and a number of customers scattered or clustered around the depot. The goal then is to find a route from the depot to the customers and back to the depot. To harden this task there are several constraints. The two constraints used in this work are the capacity constraint and the time window constraint. The capacity constraint models a maximum load for each vehicle that it cannot exceed, whereas the time window constraint adds a duration to each customer during which it expects to be served. The resulting CVRPTW can then be further categorized in a formulation with *hard Time Windows* and *soft Time Windows* respectively. In the definition of hard Time Windows, a solution is feasible if and only if each customer is reached before its due date. Contrary to soft Time Windows, where the time window constraints are relaxed and the question of feasibility is left to the designer.

Tabu Search was introduced by Fred Glover [2] and has been studied in numerous scientific papers already. The algorithm can be described basically as a "best-improvement-local-search" algorithm. It uses a single solution at a time of which it generates a number of moves leading to other solutions. These are considered to be neighbors of the current solution. From this neighborhood Tabu

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 829–836, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Search will choose the best solution that satisfies several criteria and continue by creating a new set of moves from there. Because of this behavior the run of the algorithm will leave a trajectory in the search space. To avoid undoing changes made in previous moves and to prevent the algorithm from cycling in the search space, Tabu Search stores recent moves in a memory called *tabulist*. It then forbids to make moves towards one of the configurations that had already been visited before. Over the years Tabu Search became a metaheuristic well known for its good performance when applied to the Vehicle Routing Problem (VRP) [3] [4] [5].

The paper is further organized as follows: In section 2 the problem will be presented briefly, along with a number of operators that the algorithms use to transform the problem. In section 3 we will discuss the approaches and their implementation. Results are given in section 4 and conclusions are drawn in section 5.

2 Capacitated Vehicle Routing Problem with Soft Time Windows

The basic concepts of the problem were already introduced in section 1. For a mathematical description of the problem the reader is advised to turn to the work of Thangiah [6].

To represent the problem in a computer a path oriented notation was used, specifically a string of permutated numbers representing the indices of the depot and its customers. Each tour of each vehicle starts and ends at the same depot which is assigned the index 0. Such a tour is read from left to right, so 0 is the first and last character of each permutation. In between the depot appear the indices of the customers that the vehicle visits on its tour. The whole solution is then built by concatenating all tours together into one string, overlapping at the points of intersection. For each vehicle for which no tour is assigned, a 0 is appended to the string, so an empty tour is marked by two consecutive 0s. The maximum number of vehicles R is a parameter specified by the instance of the problem.

To manipulate the solution representation several operators have been presented in the literature and many approaches made good use of them [5]. Of these operators five were selected, giving each operator the same chance to create a neighboring solution. The operators in brief are called: *Relocate*, *Exchange*, *2-opt*, *2-opt** and *Or-opt*. Relocate moves a customer from one route to another, Exchange swaps two customers of different routes. 2-opt reverses a tour or a part of it, whereas 2-opt* interchanges two tours by crossing the first half of one tour with the second half of another and vice versa. Last but not least, Or-opt moves two consecutive customers to a different place in the same tour.

These operators would stochastically sample a given number of moves from the neighborhood of all moves which is typically very large if all possibilities are to be exhausted. As an example the number of all possible neighbors using the Exchange operator is given by $\frac{|\mathcal{C}|^2-|\mathcal{C}|}{2}$. The disadvantage of stochastic sampling

is that it would be possible to miss a good move at a certain iteration and if the sampling is unable to include good moves the search would be drawn away from the better regions of the search space. To counter this a local feasibility criterion was added to each operator. This criterion disallowed to manipulate a solution when it would obviously violate the time window constraints on a local level. E.g. if a customer is to be inserted between two other customers it must be possible that this customer is reached in the best case, as well as the succeeding customer can be reached before its due date as well. This criterion was weak enough that solutions with time window violations occur and strong enough, that the algorithm could find back to a solution with all time windows satisfied.

Because the problem is attacked using a multiobjective definition there is more than one evaluation function that contributes to the fitness of an individual equally. The first evaluation function measures the total traveling distance of all vehicles combined, while the second evaluation function counts the amount of vehicles in use. Both objectives are to be minimized.

3 Algorithms

Parallel Tabu Search has been and continues to be a very active topic [7] and several strategies have been tried focusing on different levels of parallelization, i.e. functional decomposition, domain decomposition and multisearch. In the presented approaches two of them make use of functional decomposition and while a synchronous parallelization was applied early already [8], asynchronous algorithms are not yet common at this level of parallelization. Yet the asynchronous algorithms are interesting as they should perform well on both homogenous and heterogenous systems.

Multisearch spans from trivial implementations with shared memories to very complex approaches that include strategic knowledge. The third approach presented here works at this level and it was chosen to keep with a simple implementation. Section 3.3 will show a more detailed description. A combination of multisearch and functional decomposition could combine the best of two worlds.

3.1 Sequential Tabu Search

The sequential algorithm is outlined in Algorithm 1. This will be the base from which to develop the parallel algorithms. The algorithm uses three memories, the first memory is the *tabu list* and common to all Tabu Search algorithms. The tabu list is organized as a queue and will hold information about the moves made. When the tabu list is full it will forget about the oldest moves. The length of the tabu list can be specified by the *tabu tenure* parameter. The second memory is the *medium-term memory* $\mathcal{M}_{nondom}$ that keeps a list of non-dominated solutions that had been found in past neighborhoods. If the algorithm does not find "better" solutions for a certain number of iterations, it will take one of the solutions from this memory instead of generating a new neighborhood from the

current solution and continue with an emptied tabu list. Solutions are considered "better" when they dominate the current pareto front, which is stored in the third memory $\mathcal{M}_{archive}$, or that are non-dominated to the front and can be added to $\mathcal{M}_{archive}$. A chosen solution can be added to the archive when it is not dominated to the solutions in the archive and when the archive is not full. If the archive is full, the solution is added based on the result of a *crowding comparison*[9]. This comparison orders the solutions in the archive and the chosen solution by a *distance value*, which is computed by calculating the differences of the fitness values of a certain solution with respect to the other solutions. A solution that has a low distance value has similar fitness values compared to the rest of the solutions and will be deleted. This ensures that the solutions will be spread over the pareto front more equally instead of clustering at a certain position.

The initialization of the current solution is done by drawing several solutions from the powerful I1-heuristic [10] that is initialized randomly and choosing the "best" i.e. that dominates the others. The I1-heuristic is a route construction heuristic for the VRP and starts with either the customer with the earliest deadline or the one farthest away. It adds customers based on a savings value that computes the additional distance as well as time windows that the insertion of a customer will cost.

Algorithm 1. The sequential TSMO-Algorithm

```
1: procedure TSMO(iterations ↓)
2:     s ← GenerateInitialSolution()
3:     evaluations ← 0, iterations ← 0
4:     M ← InitializeMemories()
5:     while evaluations < MaximumEvaluations do
6:         if noImprovement = true then
7:             s ← SelectFrom(M_nondom ↓↑, M_archive ↓)
8:             noImprovement ← false
9:         else
10:            N ← GenerateNeighborhood(s ↓)
11:            Evaluate(N ↓↑, evaluations ↓↑)
12:            s ← Select(N ↓, M_tabulist ↓)
13:            if s ∉ N then
14:                s ← SelectFrom(M_nondom ↓↑, M_archive ↓)
15:            end if
16:        end if
17:        M ← UpdateMemories(s ↓, N ↓)
18:        if isUnchanged(M_archive ↓, iterations ↓) then
19:            noImprovement ← true
20:        end if
21:        iterations ← iterations + 1
22:    end while
23: end procedure
```

3.2 Synchronous (TSMO-SYNC) and Asynchronous (TSMO-ASYNC) Tabu Search

Common to both approaches is their parallelization of the GenerateNeighborhood() and Evaluate() functions using a master process that distributes the work among himself and several worker processes. These approaches do not improve the quality of the results, but the runtime only. In the synchronized algorithm the master divides the work into equal pieces and then waits until all the work has been collected. The asynchronous algorithm is different in that it does not wait in all cases. Sometimes the algorithm may continue with only half the neighborhood evaluated as there is a promising new solution candidate already or there are several workers waiting. The algorithm's decision function is shown in Algorithm 2. In the synchronous approach the majority of the processors are idle while the master process does the tabu check and memory update, while in the asynchronous algorithm in an ideal situation the slaves would not idle so often.

Algorithm 2. Decision function of the asynchronous TS

```
1: procedure DECISION(current ↓ 𝒩 ↓ workers ↓)
2:     c1 ← {w ∈ workers | w = waiting}
3:     c2 ← {s ∈ 𝒩 | s dominates current}
4:     c3 ← AreWeWaitingTooLong()
5:     c4 ← evaluations ≥ MaximumEvaluations
6:     return |c1| > 0 ∨ |c2| > 0 ∨ c3 ∨ c4
7: end procedure
```

3.3 Multisearch Tabu Search (TSMO-COLL)

The third approach is asynchronous and is placed in the realm of multisearch parallel algorithms. It uses a multiple points different strategies according to the classification by Crainic et al. [11]. The parameters which determine the strategy of the algorithm vary for each, but the first as they are disturbed by a random variable. The threads then work in a similar way to the sequential algorithm, but communicate improving solutions that they found along the pareto front. An improving solution here is a solution that could be added to the pareto front which is stored in $\mathcal{M}_{archive}$. Likewise a non-improving solution is one that could not be added, because it is dominated or too crowded. The communication list is initialized randomly before the main loop and different for every thread. It is used to determine the collegial searcher that will receive the next improving solution found. On the receiving end this solution is stored in $\mathcal{M}_{nondom}$ and might be considered later by the searcher, specifically if after a number of iterations it has not found better solutions and the qualities of its solution cannot compete with the one received. After the solution has been sent the communication list is rotated in that the first process is moved to the bottom. This way the communication overhead does not become too large and good solutions find their way to

other searchers who can explore this region as well, if they do not find improving solutions in the region they currently are. At the end one thread collects all the solutions of the other threads and returns the pareto front as output.

4 Results

The empirical results on a number of test problems were computed on an SGI Origin 3800 supercomputer located at the Johannes Kepler University, Linz, Austria. The Origin 3800 is a shared memory machine with 64GB RAM and consists of 128 R12000 MIPS processors, running at 400Mhz each.

The problemset used in this work is available at a website managed by Joerg Homberger[1]. The instances are grouped into 3 classes and 2 subclasses. The classes are: Clustered customer locations, Randomly distributed customer locations and a combination of both. Each class is then split into instances where the time windows are generally larger and instances where the time windows are smaller.

During the run of the algorithms solutions with constraint violations had been allowed, these solutions were excluded for the generation of the results. Only those solutions were considered that did not violate the time-window and capacity constraints.

As expected the synchronous algorithm completes faster than the sequential algorithm, its solution qualities however are not improving. This is not surprising as the behavior of the synchronous algorithm does not differ from the sequential one. As shown in Figure 4 a maximum speedup seems to be reached quickly with a few number of processors already. The formula to calculate the average speedup value is $speedup = T_s/T_p$, the mean execution time of the sequential algorithm divided by the mean execution time of the parallel algorithm.

The asynchronous algorithms like TSMO-ASYNC and TSMO-COLL are quite interesting and a combination of both could possibly provide the best of both worlds, namely faster execution time and better solution qualities. Of course the number of processors would have to increase even further and it would be even more important to examine the communication closer.

To test the statistical significance a pairwise t-test was performed on the results. Using just 3 processors the results do not show a good level of significance, but by increasing the number of processors TSMO-COLL separates from the other algorithms. With 12 processors there's a notable difference between the results of TSMO-COLL and the other algorithms that is significant at 0.1. From the numbers in Table 1 it can be seen how the algorithms perform when competing against each other.

5 Conclusions and Future Work

Improving performance in both runtime and solution quality does not require a lot of effort and where parallel systems are available researchers should make

[1] http://www.fernuni-hagen.de/WINF/touren/inhalte/probinst.htm

Table 1. Results from the SetCoverage metric [12] show that the multisearch TS is able to outperform the other two variants in terms of solution quality. Unsurprisingly TSMO and TSMO-SYNC perform equally, TSMO-ASYNC seems to perform a little bit better. These results stem from an average performance on the 400 city problems.

Algorithm	dominates	is dominated
3 processors		
TSMO	22.57%	32.78%
TSMO-SYNC	22.65%	32.51%
TSMO-ASYNC	26.09%	29.56%
TSMO-COLL	38.55%	15.00%
6 processors		
TSMO	19.53%	33.95%
TSMO-SYNC	20.31%	33.63%
TSMO-ASYNC	22.66%	30.90%
TSMO-COLL	43.91%	7.92%
12 processors		
TSMO	19.47%	34.41%
TSMO-SYNC	19.89%	33.08%
TSMO-ASYNC	20.67%	32.97%
TSMO-COLL	45.32%	4.95%

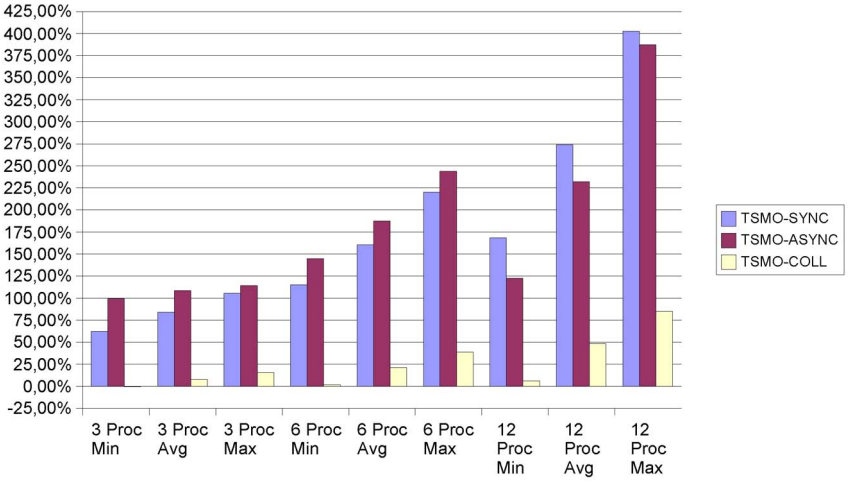


Fig. 1. Graph showing the speedup values that were calculated with respect to the sequential algorithm. Note that TSMO-ASYNC can perform better with up to 6 processors and then falls behind TSMO-SYNC. This may be due to the way the master in the asynchronous variant splits the work. Using more processors it could be beneficial for the master not to be involved in doing evaluation for itself, but function purely as a control thread. TSMO-COLL can achieve a little speedup due to the parallelization of the initialization phase before the main loop of the algorithm. Each process in TSMO-COLL just generates one solution from the I1-heuristic.

use of the extra processing power. An interesting approach is the asynchronous master slave algorithm as it performed quite well, yet achieved a better speedup up to 6 processors. Another point to consider is the deployment in a heterogeneous computing environment where the asynchronous variant is having quite an advantage. Whether the collaborative TS in this simple form delivers a good tradeoff between extra solution quality and runtime performance is questionable, however the results from the metric suggests that solutions found were more robust than just the master-slave variants of the algorithm.

What remains for the future would be a comparison between the TSMO versions here and the well established multiobjective evolutionary algorithms in both runtime and solution quality and on different problems, as well as combining the multisearch TS with the asynchronous TS to get the best of both worlds.

References

1. Dantzig, G.B., Ramser, R.H.: The Truck Dispatching Problem. Management Science 6, 80–91 (1959)
2. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers & Operations Research 13, 533–549 (1986)
3. Cordeau, J.F., Laporte, G.: Tabu Search Heuristics for the Vehicle Routing Problem. GERAD Technical report G-2002-15, University of Montreal, Canada (2002)
4. Bräysy, O., Gendreau, M.: Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. Transportation Science 39(1), 104–118 (2005)
5. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, Part II: Metaheuristics. Transportation Science 39(1), 119–139 (2005)
6. Thangiah, S.R.: Vehicle Routing with Time Windows using Genetic Algorithms. The Practical Handbook of Genetic Algorithms: New Frontiers, pp. 253–278. CRC Press, Boca Raton, USA (1995)
7. Crainic, T.G., Gendreau, M., Potvin, J.Y.: Parallel Tabu Search. Parallel Metaheuristics: A New Class of Algorithms, pp. 289–314. John Wiley & Sons, Chichester (2005)
8. Garcia, B.L., Potvin, J.Y., Rousseau, J.M.: A Parallel Implementation of the Tabu Search Heuristic. Computers & Operations Research 21(9), 1025–1033 (1994)
9. Deb, K., Agrawal, S., Prata, A.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In: Proceedings of the Parallel Problem Solving from Nature VI Conference, pp. 849–858 (2000)
10. Solomon, M.M.: Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints. Operations Research 35, 254–265 (1987)
11. Crainic, T.G., Toulouse, M., Gendreau, M.: Towards a Taxonomy of Parallel Tabu Search Algorithms. INFORMS Journal on Computing 9(1), 61–72 (1997)
12. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. Doctoral thesis, ETH No:13398, Swiss Federal Institute of Technology, Zurich, Switzerland (1999)

Bandit-Based Monte-Carlo Planning for the Single-Machine Total Weighted Tardiness Scheduling Problem

Gabriel Kronberger and Roland Braune

Upper Austria University of Applied Sciences
Research Center Hagenberg
Softwarepark 11, 4232 Hagenberg, Austria
{gabriel, roland}@heuristiclab.com

Abstract. The balance of exploration and exploitation is the essence of any successful meta-heuristic. The *Multi-armed Bandit Problem* represents a simple form of this general dilemma. This paper describes two heuristic optimization methods that use a simple yet efficient allocation strategy for the bandit problem called *UCB1* to control the optimization process.

The algorithms are applied to the well known *Single Machine Total Weighted Tardiness Problem* and the results compared to the results of other successful meta-heuristics for this scheduling problem.

1 Introduction

1.1 Motivation

One of the central features of every efficient meta-heuristic is the balance between exploration of the solution space and exploitation of subareas where comparatively better solutions have been found. This problem is represented in simplified form in the general *Multi-Armed Bandit Problem*. The Multi-armed Bandit Problem describes the situation of an unfair casino with multiple one-armed bandit gambling machines (or one multi-armed bandit) that each have a different rate of winning plays and possibly unequal rewards. The goal for a player is to allocate plays to the machines in such a way as to maximize the expected total reward even though in the beginning it is unknown how the rewards of each machine are distributed.

Likewise in heuristic optimization the landscape of the solution space is initially unknown. However as the optimization progresses and more and more solutions are generated it is possible and necessary to concentrate the computational effort to the subareas of the solution space that surround the best solutions found so far while not totally ignoring unexplored areas. The idea followed in this paper is to use an effective allocation strategy for the Bandit Problem to control a heuristic optimization algorithm.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 837–844, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Two algorithm variants based on a specific allocation strategy for the bandit problem are presented in this paper. Both variants are applied to the *Single-Machine Total Weighted Tardiness Problem* and tested with a well known benchmark set for this scheduling problem. The benchmark results are briefly compared to the results of the best performing methods for the SMTWTP.

1.2 The Multi-armed Bandit Problem

The K-armed Bandit Problem can be described as K random variables with $X_i (0 \leq i < K)$. Where X_i is the stochastic reward given by the gambling machine with index i. The distributions of X_i are independent. However the distributions are generally not identical. The laws of the distributions and the expectancies μ_i for the rewards X_i are unknown.

The goal is to find an allocation strategy that determines the next machine to play based on past plays and received rewards that maximizes the expected total reward for the player. Or put in other words: to find a way of playing that minimizes the regret which is defined as the expected loss occurred by not playing the optimal machine each time.

Since a player doesn't know which of the machines is the best he can only make assumptions based on previous plays and received rewards and concentrate his efforts on the machines that gave the highest rewards in the past. However it is necessary to strike a good balance between exploiting the currently best machine and exploring the other machines to make sure that none of those is even better.

1.3 Allocation Strategy for the Bandit Problem: UCB1

Auer et al. [1] describe a number of simple and efficient strategies that achieve logarithmic regret uniformly over time. This is an improvement over other allocation strategies like the ϵ-greedy strategy that achieve logarithmic regret only asymptotically. Figure 1 shows the definition of the *UCB1* allocation strategy as given by Auer et al.

Based on the UCB1 Kocsis and Szepesvári [8] formulate a rollout-based Monte-Carlo planning algorithm *UCT* (Upper Confidence Bounds applied to Trees). The

Strategy: UCB1
Initialization: Play each machine once.
Loop: Play machine j that maximizes $\overline{x}_j + \sqrt{\frac{2 \ln n}{n_j}}$, where $\overline{x}_j$ is the average reward obtained from machine j, n_j is the number of times machine j has been played so far and n is the overall number of plays done so far.

Fig. 1. UCB1 as described by Auer et al. [1]

algorithm builds a search tree in memory by repeatedly expanding one of the leaf nodes. In each iteration the algorithm starts at the root node and selects a path through the internal nodes of the tree to one of the leaves. In each node the algorithm chooses the following node based on UCB1. When the algorithm reaches one of the leaf nodes it expands the tree by creating new leaf nodes and executes Monte-Carlo evaluations for the leaves. The results of the simulations are then used to update the evaluations of the visited nodes by backtracking the path to the root of the tree. The benefit of using UCB1 is that the algorithm searches the most beneficial branches of the tree very deeply while still trying unexplored branches from time to time. [8] demonstrates that UCT performs well when applied to the stochastic shortest path problem and to random (P-game) trees.

The algorithm is currently the best performing method for 9x9 computer Go programs. Further improved and augmented variants of UCT are used in many strong Monte-Carlo based Go programs like for example MoGo [7].

1.4 Single-Machine Total Weighted Tardiness Problem

The SMTWTP can be stated as follows:

A set of n jobs has to be scheduled on a single machine. The machine can only process one job at a time and the execution of a job cannot be interrupted. Each job j becomes available at time zero, requires a predefined processing time p_j, has a positive weight w_j and a due date d_j. A schedule is constructed by sequencing the jobs in a certain order such that the completion time C_j of each job can be computed. If the completion time exceeds a job's due date, the objective function gets increased by a tardiness penalty $w_j T_j$, where $T_j = \max(0, C_j - d_j)$. The optimization goal is to find a processing order which minimizes the value of the sum (1).

$$\sum_{j=1}^{n} w_j T_j \tag{1}$$

The list of best-performing approaches for the SMTWTP includes Iterated Dynasearch [6], an enhanced Tabu Search [5] and a specialized Ant Colony System [4].

2 Bandit-Based Heuristic Optimization

In this section two variants of Bandit-based optimization algorithms for the SMTWTP are described. The first variant *UCB-SMSP-Matrix* uses a matrix representation of jobs and positions that is similar to the representation used in ACO Algorithms. The second variant *UCB-SMSP-Tree* is an adaption of *UCT* for the SMTWTP.

2.1 UCB-SMSP-Matrix

The algorithm works on a matrix of average rewards represented by the tuple *(reward_sum, tries)*. Each row holds represents a position in the job-sequence

and each column represents one of the jobs. This representation is very similar to the pheromone matrix used in ACO algorithms. Each tuple $t_{i,j}$ of the matrix holds the the number of solutions generated with job j in position i and the sum of rewards received for these solutions. Initially all tuples are set to $(0, 0)$.

The algorithm iteratively generates new solutions and then uses the weighted tardiness of the solution to update the average rewards until it is terminated. Solutions are generated by selecting one of the open jobs for each position based on UCB1. The reward is the relative quality of the current solution compared to the best solution found by the algorithm so far. To reduce a possible bias the algorithm fills the positions of the sequence in a random order. Algorithm 1 shows the pseudo code of UCB-SMSP-Matrix.

```
Data: T := (t_{i,j})_{N_Jobs×N_Jobs}, t_{i,j} := (reward_sum,tries);
S_cur;
Result: S_best
initMatrix();
while !terminationCriterion do
    permutation ← shufflePositionPermutation(permutation);
    foreach i = 1 .. N_Jobs do
        position ← permutation[i];
        foreach job ∈ unscheduled jobs do
            v_job ← calculateUcb1(t_{position,job});
        end
        job_best ← select job with largest v_job;
        S_{cur,position} ← job_best;
        if weightedTardiness(S_cur) < weightedTardiness(S_best) then
            S_best ← S_cur;
        end
    end
    foreach position do
        job ← S_{cur,position};
        reward ← ( weightedTardiness(S_best) / weightedTardiness(S_cur) );
        t_{position,job} ← t_{position,job} + (reward, 1);
    end
end
```

Algorithm 1. Pseudo code of UCB-SMSP-Matrix

2.2 UCB-SMSP-Tree

UCB-SMSP-Tree is basically a UCT variant for single machine scheduling problems. It iteratively builds a search tree of partial solutions starting from an empty sequence. The sum of rewards and the number of tries are stored in the nodes of the tree instead of a matrix of nodes. Partial solutions are built in a linear fashion starting with the first position. At the leaves of the search tree the partial solution is completed by scheduling the remaining jobs at random

```
Data: N :=(job, reward_sum, tries);
S_cur;
Result: S_best
N_root ← createNode();
while !terminationCriterion do
    N_cur ← N_root;
    S_cur ←createEmptySequence();
    while tries of N_cur > 0 do
        if children(N_cur) = ∅ then
            createChildren(N_cur);
        end
        foreach N_child ←children(N_cur) do
            v_child ←calculateUcb1(N_child);
        end
        N_best ← select the child N_child with largest v_child;
        S_cur,position ← N_best;
        N_cur ← N_best;
    end
    S_cur ← completeRandom(S_cur);
    if weightedTardiness(S_cur) < weightedTardiness(S_best) then
        S_best ← S_cur;
    end
    foreach N_cur ← S_cur do
        reward ← weightedTardiness(S_best) / weightedTardiness(S_cur);
        N_cur ← N_cur + (0, reward, 1);
    end
end
```

Algorithm 2. Pseudo code of UCB-SMSP-Tree

positions. The resulting solution is then evaluated and the weighted tardiness is then used to update the tree nodes while backtracking the path through the tree. Algorithm 2 shows the pseudo code of UCB-SMSP-Tree.

2.3 Hybrid Algorithms: Adding Local Neighborhood Search

Experiments with both algorithm variants show that it's hopeless to find the best known solutions in reasonable runtime with such a simple approach. To find close to optimal solutions it is necessary to introduce a second optimization phase. In the second optimization phase a greedy steepest descent algorithm is used to locally improve the solution generated in the first phase.

Typical neighbourhoods for scheduling problems are the *swap* (*2-opt*) and the *move* neighbourhood. The swap neighbourhood is the set of solutions that can be obtained by interchanging the positions of two arbitrary jobs. The *move* neighbourhood is the set of solutions that can be obtained by moving one job from position i to position j and shifting the jobs on positions between i and j in the appropriate direction to fill the empty position. The efficiency improvements

described in [6] were implemented to improve the runtime of the neighbourhood search.

Table 1 shows the benchmark results of local search in these two neighbourhoods as well as in the combined neighbourhoods *swap+move* and *move+swap* for 100 random restarts. Combining the neighbourhoods means the algorithm runs a steepest descent in the first neighbourhood and when it reaches a local optimum it tries to improve the solution further by continuing the steepest descent in the second neighbourhood. The results coincide with the results of [4] that showed that the combined neighbourhoods generally lead to better solutions than each neighbourhood alone while the average runtime of the search in the combined neighbourhood is only slightly larger than searching only the in the first neighbourhood.

Two problem sets for the SMSP are used in this paper. The first set is the 100-job SMSP benchmark set from the OR-Library[1] [3] containing 125 problem instances. The second benchmark set also contains 125 instances but for 200 jobs. It is the same benchmark set as used in [2] which was kindly given to us by Prof. Selim Akturk.

The columns given in the results tables are:

n_{best}: Is the median over all trials for the number of problem instances for which the best known solutions is found (max. 125).
t_{avg}: Average computing time over all problems.
t_{max}: Maximal computing time over all problems.
Δ_{avg}: Average relative percent difference over all problems.
Δ_{max}: Maximal relative percent difference over all problems.

Table 1. Performance of steepest descent neighbourhood search. 100 random restarts for each of the 125 problem instances from both benchmark sets.

Neighbourhood	n_{best}	Δ_{avg}	$t_{avg}[ms]$	$t_{max}[ms]$
100 jobs				
swap	21	11.28%	38	282
move	26	6.15%	26	407
swap+move	36	3.35%	44	329
move+swap	34	2.47%	93	532
200 jobs				
swap	21	6.12%	348	1670
move	24	9.67%	1336	4609
swap+move	26	2.91%	566	6250
move+swap	26	2.32%	1338	3578

Based on this results both combined neighbourhoods are used to locally improve the generated solutions before updating the rewards. Like in [4] the swap+move neighbourhood is used for one half of the generated solutions and

[1] `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`

the move+swap neighbourhood for the other half. At initialization a reasonably good starting solution is generated with the *Apparent Urgency* construction heuristic [9] which is locally improved in both combined neighbourhoods.

The introduction of local search improves the performance of the described algorithms significantly. However after this change it is also necessary to adjust the reward-update function accordingly because the improved sequence is different from the sequence sampled in the first optimization phase. The adjusted update function increases the tries of the nodes visited in the first phase and adds the rewards to the nodes that represent the sequence after the local improvement. Another improvement is the use of the tuned variant of UCB1 as given in [1].

3 Results

All experiments were executed on a single core workstation with a 3GHz Intel Pentium 4 and 1GB of RAM. Implementation language is Java version 1.6. Table 2 shows the results of both algorithms and both benchmark sets. The column *trials* gives the number of independent runs for each of the 125 problem instances from both sets.

When applied to the 100-job benchmark set the algorithm was terminated as soon as the best known solutions were found. The maximum runtime for each instance was set to 60 seconds. When applied to the 200-job benchmark we fixed the runtime to 180 seconds for each instance because we were interested if and for how many problems the algorithms would be able to find new best solutions.

Column n_{best} gives the median of the number of problem instances for which the best known solution was found. The number in brackets is the median of the number of problem instances for which the algorithm could improve the best known solution. Over all runs we were able to find new best solutions for 22 of the 125 problem instances.

Table 2. Results of the algorithms applied to the 100-job benchmark set from OR-Library and the 200-job benchmark set from [2]

Algorithm	trials	n_{best}	Δ_{avg}	Δ_{max}	$t_{avg}[s]$	$t_{max}[s]$
100 jobs						
UCB-SMSP-Tree	100	122	0.00083%	0.214%	11.8	60
UCB-SMSP-Matrix	100	124	0.00025%	0.111%	3.8	60
200 jobs						
UCB-SMSP-Tree	10	63 [6]	0.0243%	0.61%	180	180
UCB-SMSP-Matrix	10	83 [12]	0.0895%	10.05%	180	180

4 Conclusion and Future Research

The results show that the performance of both bandit-based algorithms is reasonably good. However they can't surpass the performance of specialized methods

for the SMTWTP. Iterated Dynasearch [6], Ant Colony System [4] and Tabu Search [5] are capable of finding the best known solution for each instance of the 100-job benchmark set from the OR-Library. On the other hand our algorithms and parameter settings still leave room for further improvements.

Anyway, the SMTWTP is a static scheduling problem and is therefore not a perfect testbed for bandit-based optimization algorithms. Applying bandit-based algorithms to dynamic production planning problems remains an open topic for future research.

References

1. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time Analysis of the Multiarmed Bandit Problem. Machine Learning 47(2-3), 235–256 (2002)
2. Avci, S., Selim Akturk, M., Storer, R.H.: A problem space algorithm for single machine weighted tardiness problems. IIE Transactions 35, 479–486 (2003)
3. Beasley, J.E.: OR-Library: Distributing Test Problems by Electronic Mail. Journal of the Operational Research Society 41(11), 1069–1072 (1990)
4. den Besten, M., Stützle, T., Dorigo, M.: Ant Colony Optimization for the Total Weighted Tardiness Problem. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) Parallel Problem Solving from Nature-PPSN VI. LNCS, vol. 1917, pp. 611–620. Springer, Heidelberg (2000)
5. Bilge, Ü., Kurtulan, M., Kirac, F.: A tabu search algorithm for the single machine total weighted tardiness problem. European Journal for Operational Research 176(3), 1423–1435 (2007)
6. Congram, R.K., Potts, C.N., van de Velde, S.N.: An Iterated Dynasearch Algorithm for the Single-Machine Total Weighted Tardiness Scheduling Problem. INFORMS Journal on Computing 14(1), 52–67 (2002)
7. Gelly, S., Wang, Y., Munos, R., Teytaud, O.: Modifications of UCT with Patterns in Monte-Carlo Go. Technical Report, No. 6062. INRIA (2006)
8. Kocsis, L., Szepesvári, C.: Bandit based Monte-Carlo Planning. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 282–293. Springer, Heidelberg (2006)
9. Morton, T.E., Rachamadugu, R.M., Vepsalainen, A.: Accurate myoptic heuristics for tardiness scheduling. GSIA Working Paper No. 36–83–84. Carnegie-Mellon University Pittsburgh, PA (1984)

Using GAs to Obtain an Optimal Set of Codes for an Ultrasonic Local Positioning System

Fernando J. Álvarez–Franco, Horacio M. González–Velasco,
Carlos J. García–Orellana, Miguel Macías–Macías,
and Ramón Gallardo–Caballero

CAPI Research Group, University of Extremadura
Polytechnic School. Av. de la Universidad, s/n. 10071 Cáceres - Spain
fernando@capi.unex.es
http://capi.unex.es

Abstract. Signal coding and pulse compression techniques have been recently introduced in Local Positioning Systems as a means to enhance the measurement precision of these systems and to increase their operation frequency. This work presents a Genetic Algorithm that performs the search for an optimal family of binary codes, to be used in a system based on ultrasonic technology. The developed algorithm takes into account the transduction effect of the emitters on the correlation properties of the modulated family to obtain a set of codes that exhibits a superior performance than other families previously used.

1 Introduction

In the last years, Local Positioning Systems (LPS) have become very popular, not only in the field of mobile robotics, where they constitute a navigation aid for autonomous vehicles [1], but also in applications of different nature, such as the location of persons in the rooms of an office building [2], or the location of findings in an archaeological site [3].

Among the different technologies commonly used for LPS - infrared, RF and ultrasonics - the last one offers a low cost solution accurately enough for most applications. However, in a classical system, where ultrasonic signals are detected by envelope thresholding, there are mainly three disadvantages to be considered. First, the maximum precision obtained in the determination of the position is limited to several centimeters in best cases. Second, high frequency sounds such as key jangling or rustling paper can interfere with the ultrasonic signals and affect the proper performance of the system. Finally, the maximum operation frequency is limited by the critical time that is necessary to wait between consecutive emissions from different beacons, in order to have all the echoes from the first emission attenuated far below the detection threshold. The first two disadvantages can be overcome by incorporating signal coding and pulse compression techniques from radar theory. These techniques have been widely used in advanced airborne sonars since the nineties [4] [5], providing these systems with a sub-millimetric precision and a high robustness to noise, at the expense of increasing the computational complexity of the signal processing tasks. With respect to the operation frequency, an

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 845–852, 2007.
© Springer-Verlag Berlin Heidelberg 2007

appropriate selection of the codes to be emitted allows the simultaneous emission of all the beacons minimizing the cross-talk interference between them – multi-mode operation –. In this case, it is not necessary to wait a critical time between emissions from different beacons, as the signal coming from every beacon can be unmistakably recognized by the receiver.

This work presents a Genetic Algorithm (GA) that searches for a family of codes with a minimum interference between them when matched filtering detection is carried out in the receiver. The rest of the paper is organized as follows: Section 2 provides a general view of the basics of ultrasonic local positioning. Section 3 explains in more detail the improvements achieved with the introduction of signal coding and pulse compression techniques. Section 4 describes the GA proposed in this work, whose results are presented in Section 5. Finally, some conclusions are outlined in Section 6.

2 Ultrasonic Local Positioning

The principle of operation of an ultrasonic LPS is rather simple, and can be easily explained with the aid of Fig. 1. As can be seen, these systems are composed of several beacons placed at known positions in the environment, and these beacons emit ultrasonic signals that are detected by a receiver placed in the object to be located. Then, the receiver computes the times-of-arrival (TOA) of these signals, and multiplies these times by the speed of sound in air c to obtain the distances to the beacons $d = TOA \cdot c$. The 3D position of the object (x, y, z) can be calculated with a minimum of three beacons, by solving the following system of equations:

$$\sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} = d_k \quad \text{with} \quad k = \{1, 2, 3\} \ . \tag{1}$$

where (x_k, y_k, z_k) represents the position of the k-th beacon. This method is known as trilateration or spherical positioning, since (1) are the equations of three spheres centered in the position of the active beacons. Trilateration is the simplest way to perform relative positioning, and there are some notable works based on this technique [2][6]. However, the main drawback of this method is the need of an RF synchronization signal, as the receiver must know the precise moment in which the beacons start their emissions. This need increases the price of the system, and one of the strong points of using ultrasonic technology is lost.

The synchronization signal can be removed by adding a new beacon, named master beacon, and computing the time-differences-of-arrival (TDOA) between the signals coming from the other three beacons and the signal coming from the master, instead of computing the absolute TOAs. This method is known as multilateration or hyperbolic positioning and, in this case, the 3D position of the object can be determined through the following system of equations:

$$\sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} - \sqrt{(x_m - x)^2 + (y_m - y)^2 + (z_m - z)^2} = c \cdot TDOA_k \quad \text{with} \quad k = \{1, 2, 3\} \ . \tag{2}$$

where (x_k, y_k, z_k) is again the position of the k-th beacon, (x_m, y_m, z_m) is the position of the master beacon, and $TDOA_k$ represents the difference between the

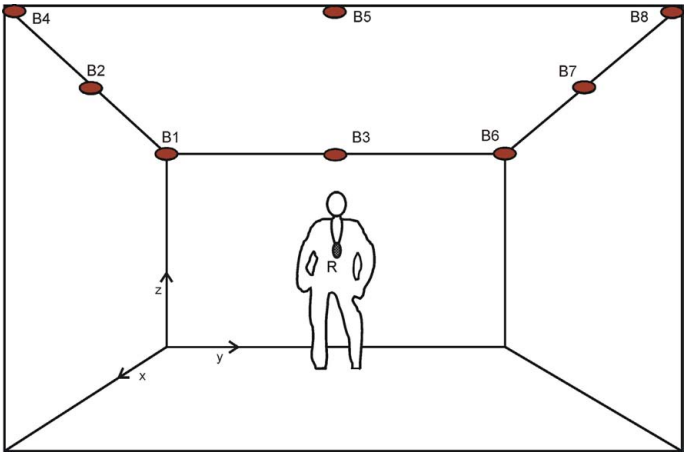


Fig. 1. Ultrasonic LPS with one receiver (R) and 8 beacons (B1-B8)

TOA of the signals coming from the k-th beacon and the TOA measured for the master beacon. In these equations, the sound speed c is a known parameter that must be estimated with a thermo-hygro sensor. Nevertheless, this estimation can be avoided by adding a new beacon to introduce another equation in the system (2) and treat c as a new unknown. This means that five is the minimum number of beacons necessary to perform relative positioning with an ultrasonic LPS without the need of a synchronization signal and a measured value of c. Most systems incorporate more beacons to add redundancy to the measurements and to avoid shadow areas. Thus, a total number of eight beacons is typical in some systems already developed [3][7].

3 Enhancing the Performance: Pulse Compression and Signal Coding

As stated in the introduction, the main inconvenient of ultrasonic ranging modules, the central element of an ultrasonic LPS, is the low accuracy of these modules, typically in the order of some cms or even dms. In a classical ranging module, the emitter transmits an ultrasonic burst which is rectified and integrated at the receiver. The TOA is computed when the integrated signal exceeds a detection threshold that must be chosen to avoid the validation of echoes and noise. Although these modules usually incorporate an automatic control gain that compensates for the increasing attenuation of the signal with traveling distance, it is clear that they cannot be very precise, since they are basically detecting the envelope of the emitted signal. Moreover, if some measurements are required in a short period of time, as it is the case of an LPS, classic ranging modules present another problem: as signals coming from different emitters cannot be distinguished by the receiver, the emissions must be

performed sequentially, thus limiting the maximum operation frequency of the system. This might be a critical parameter when trying to locate moving objects.

Some recent ultrasonic LPS have incorporated pulse compression techniques to surmount these problems [7][8]. In these systems, the emitter does not transmit an ultrasonic burst but a binary code that has been previously modulated to adapt its spectrum to the bandwidth of the transducer. In the receiver, this signal is digitized and correlated with the emitted pattern to gather all its energy in a single peak. When this peak exceeds the detection threshold, the time of arrival is computed. This process gives the system the precision that can be achieved with the emission of very short pulses keeping, however, the high signal-to-noise ratio provided by the emission of long pulses (and hence the name *pulse compression*).

Furthermore, if the codes are chosen with low values of cross-correlation between them, they can be simultaneously emitted. In this case, the receiver incorporates a bank of as many matched filters as different codes are emitted. As a result, it is capable to distinguish between different beacons, and the frequency of operation of the system can be notably increased. Many families of binary codes with good correlation properties have been proposed and thoroughly studied, such as Golay pairs [9], Gold sequences [10], Kasami sequences [11] or complementary sets of sequences [12]. The search for an optimal family is still an active line of research in the field of signal processing, as demonstrated by the continuous appearance of new families: LS codes [13] and Ear-ZCZ codes [14] are recent examples.

4 Search for an Optimal Family of Codes. Proposed GA

The main goal of this work is to find an optimal family of codes for an ultrasonic LPS. This family must contain 8 codes that will be assigned to different beacons. These codes are BPSK modulated with a $50\,kHz$ ultrasonic carrier, giving a total emission time of $t_e = 20\,\mu s \times L$, where L is the number of bits in the codes.

The Figure of Merit commonly used to quantify the goodness of performance of a family with M codes of L bits, say $\{c_i[n];\, 0 \leq n < L,\, 0 < i \leq M\}$, is the bound B, defined as $B = \frac{1}{L} \cdot \max(\theta_A, \theta_C)$, where θ_A stands for the maximum sidepeak obtained in all the autocorrelations and θ_C is the maximum value obtained in the crosscorrelations between all the sequences contained in the family, i.e.:

$$\theta_A = \max\left\{\phi_{cici}\left[k\right];\, \forall i \in \left[1, \ldots, M\right],\, \forall k \neq 0\right\} \quad . \tag{3a}$$

$$\theta_C = \max\left\{\phi_{cicj}\left[k\right];\, \forall i, j \in \left[1, \ldots, M\right],\, i \neq j,\, \forall k\right\} \quad . \tag{3b}$$

However, this is a very poor parameter which does not take into account the possible contribution of all the codes in a multiple reception. A more appropriate one is the sum of all the maximum sidepeaks obtained in the autocorrelations and all the maximum values obtained in the crosscorrelations, normalized by the product between the number of bits L and the number of codes in the family M:

$$F_1 = \frac{1}{M \cdot L} \sum_{i=1}^{M} \max \{\phi_{cici}\,[k]\,; k \neq 0\} + \frac{1}{M \cdot L} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} \max \{\phi_{cicj}\,[k]\,; \forall k\} \quad . \quad (4)$$

This function has $M \cdot L$ discrete variables and thus analytical methods based on gradient evaluation cannot be applied to search for its minimum. An exhaustive search is not possible either, since there are $2^{M \cdot L}$ families with M codes of L bits. Here we propose a Genetic Algorithm (GA) that performs the search for this minimum, taking also into account the effects that modulation and transducer filtering have on the correlation properties of the family. Therefore, the objective function of our algorithm is a slightly different version of (4):

$$F_2 = \frac{1}{M \cdot \phi_0} \sum_{i=1}^{M} \max \{\phi_{sipi}\,[k]\,; k \notin [-N+1, N-1]\} + \frac{1}{M \cdot \phi_0} \sum_{i=1}^{M} \sum_{\substack{j=1 \\ j \neq i}}^{M} \max \{\phi_{sipj}\,[k]\,; \forall k\} \quad . \quad (5)$$

where p_i –correlation pattern– represents the ideally modulated code c_i , s_i –filtered signal– is the signal resulting from the emission of p_i through the transducer, $\phi_0 = \phi_{pipi}[0]$ is the autocorrelation peak of any p_i, and N is the number of samples in the modulation symbol.

The GA is initialized with a population of 50 families of M codes randomly generated. Every new generation, the objective function (5) is evaluated for all the families and these values are used to linearly rank the families into the fitness interval $[0.6 - 1.4]$. Then, the best adapted families (20% of the total) survive and the rest are selected to generate the offspring through a Roulette Wheel Selection method. Selected families are combined by means of a single-point crossover operator, with a crossover probability of 0.7. Also a mutation operator is applied with a mutation probability of 0.02.

5 Results

The algorithm presented in the previous section has been programmed to find a family with 8 codes of 127 bits, identical in size to the set of Gold codes used in [7]. The evolution of the best adapted individual and the average objective value are shown in Fig. 2 for 300 generations, together with the hexadecimal representation of the achieved solution. To obtain the binary representation of these codes, the first digit must be expressed as a group of only 3 bits (note that this digit is never greater than 7).

The bound of the family shown in Fig. 2, $B_{GA} = 0.2677$, is slightly worse than the bound of the Gold set $B_{Gold} = 0.2126$, but its objective value is considerably

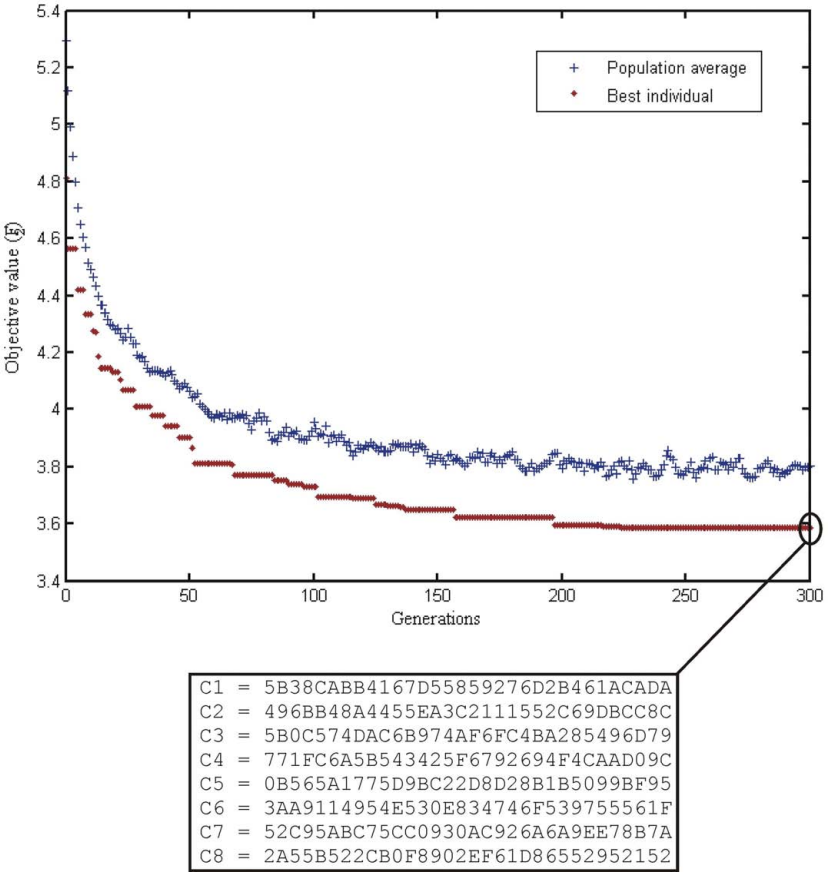


Fig. 2. GA evolution and best adapted family

better: $F_{2,GA} = 3.5866$ against $F_{2,Gold} = 4.8180$. Figure 3 simulates the operation of the LPS depicted in Fig. 1 when the dimensions of the rectangular room are $x = 3\,m$, $y = 4\,m$ and $z = 2.5\,m$. The receiver R is placed in the position $(0, 0, 1.5)$ – one meter below beacon 1 on the same vertical –, and all the beacons emit their signals simultaneously. First graph in Fig. 3 shows the signal acquired by the receiver for $20\,ms$, enough time for all the codes to be completely received in this position. The rest of the graphs show the correlation of this signal with the eight patterns, carried out to detect the arrival of the corresponding codes. As can be seen, in all cases a peak is obtained in the precise moment in which the code matched with the correlation pattern reaches the receiver, and this happens despite these codes are being received partially overlapped. If the first peak appearing in this figure is considered the temporal reference, any three of the remaining peaks can be used to obtain a set of TDOAs and compute the position of the receiver through (2).

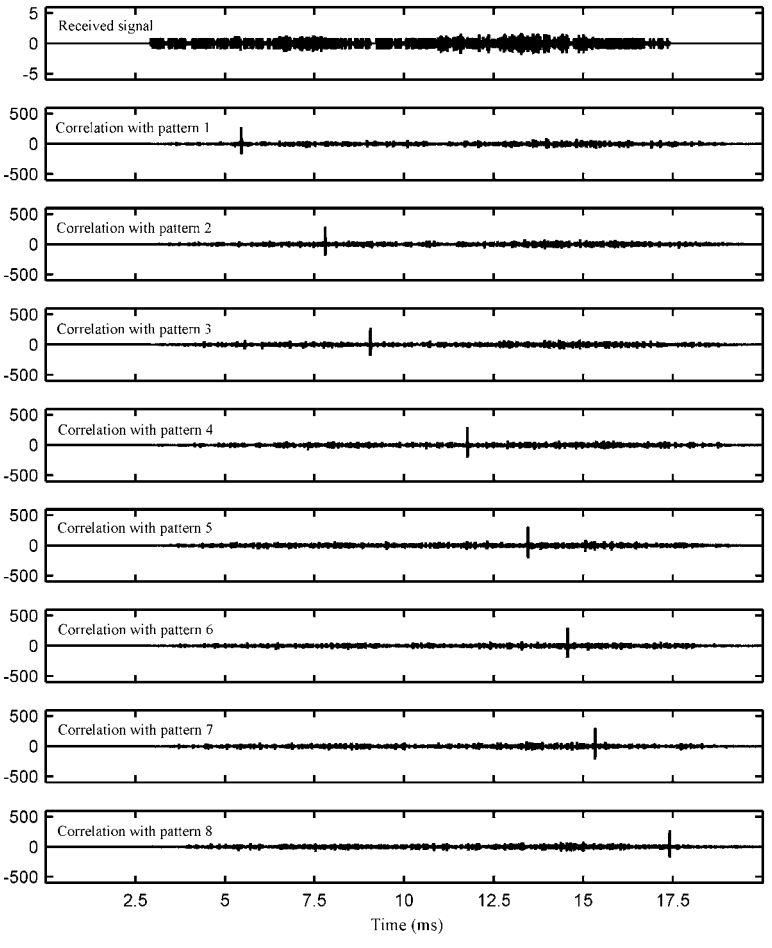


Fig. 3. Simulation of the LPS system depicted in Fig. 1

6 Conclusions

This work has presented a new GA that obtains an optimal family of codes for an ultrasonic Local Positioning System. This family can be used to encode the signals emitted by the system, allowing their simultaneous transmission with a minimum interference between them when matched filtering is performed in the receiver. It has been shown that the GA is capable to find a binary family whose performance is superior than that of the well-known Gold set for a particular LPS, since this algorithm takes into account the negative effect that the transducers have on the correlation properties of the modulated codes.

Finally, it is important to remark that, although in this work the GA has been used to find an optimal family with an specific size, the code of this algorithm is

completely parameterized, and thus it can be used to search for optimal families containing an arbitrary number of codes of any length.

Acknowledgements

This work has been partially supported by the Regional Government of Extremadura through project PRI06A227.

References

1. Aoyagi, S., Tacaño, M., Noto, H.: Development of indoor mobile robot navigation system using ultrasonic sensors. In: ICV 1998. Proc. of the IFAC Workshop on Intelligent Components for Vehicles, Seville, Spain, pp. 345–349 (1998)
2. Addlesee, M., Curwen, R., Hodges, S., Newmann, J., Steggles, P., Ward, A., Hopper, A.: Implementing a sentient computing system. IEEE Computer 34, 50–56 (2001)
3. Jiménez, A.R., Seco, F.: Precise localisation of archaeological findings with a new ultrasonic 3d positioning sensor. Sensors and Actuators 123, 214–233 (2005)
4. Peremans, H., Audenaert, K., Campenhout, J.V.: A high resolution sensor based on tri-aural perception. IEEE Transactions on Robotics and Automation 9, 36–48 (1993)
5. Jörg, K.W., Berg, M.: Sophisticated mobile robot sonar sensing with pseudo-random codes. Robotics and Autonomous Systems 25, 241–251 (1998)
6. Randell, C., Muller, H.: Low cost indoor positioning system. In: Proc. of the 3rd International Conference on Ubiquitous Computing, Atlanta, USA, pp. 42–48 (2001)
7. Villadangos, J.M., Ureña, J., Mazo, M., Hernández, A., Álvarez, F., García, J.J., Marziani, C., Alonso, D.: Improvement of ultrasonic beacon-based local position system using multi-acces techniques. In: Proc. of IEEE International Symposium on Intelligent Signal Processing, Faro, Portugal (2005)
8. Hazas, M., Ward, A.: A high performance privacy-oriented location system. In: Proc. of the 1st IEEE International Conference on Pervasive Computing and Communications (PerCom 2003), pp. 216–223. IEEE Computer Society Press, Los Alamitos (2003)
9. Golay, M.J.: Complementary series. IRE Transactions of Information Theory IT-7, 82–87 (1961)
10. Gold, R.: Optimal binary sequences for spread spectrum multiplexing. IEEE Transactions of Information Theory IT-13, 619–621 (1967)
11. Kasami, T.: Weight distribution formula for some class of cyclic codes. Technical Report R–285, Coordinated Science Lab. University of Illinois (1968)
12. Tseng, C.C., Liu, C.L.: Complementary sets of sequences. IEEE Transactions on Information Theory IT-18, 644–652 (1972)
13. Li, D.: The perspective of large area synchronous cdma technology for the fourth generation mobile radio. IEEE Commun. Mag 41, 114–118 (2003)
14. Zhang, X.L., Hatori, M.: New sequence pairs with ear zero correlation windows. In: Proc. of the IEEE International Conference on Communications, pp. 3261–3264. IEEE Computer Society Press, Los Alamitos (2004)

Using Omnidirectional BTS and Different Evolutionary Approaches to Solve the RND Problem

Miguel A. Vega-Rodríguez[1], Juan A. Gómez-Pulido[1], Enrique Alba[2], David Vega-Pérez[1], Silvio Priem-Mendes[3], and Guillermo Molina[2]

[1] Dep. Techn. of Computers and Communications, Univ. of Extremadura, Caceres, Spain
{mavega, jangomez}@unex.es, dvperez@indra.es
[2] Dep. of Computer Science, Univ. of Malaga, Malaga, Spain
{eat, guillermo}@lcc.uma.es
[3] Polytechnic Institute of Leiria, School of Technology and Management, Leiria, Portugal
smendes@estg.ipleiria.pt

Abstract. RND (Radio Network Design) is an important problem in mobile telecommunications (for example in mobile/cellular telephony), being also relevant in the rising area of sensor networks. This problem consists in covering a certain geographical area by using the smallest number of radio antennas achieving the biggest cover rate. To date, several radio antenna models have been used: square coverage antennas, omnidirectional antennas that cover a circular area, etc. In this work we use omnidirectional antennas. On the other hand, RND is an NP-hard problem; therefore its solution by means of evolutionary algorithms is appropriate. In this work we study different evolutionary approaches to tackle this problem. PBIL (Population-Based Incremental Learning) is based on genetic algorithms and competitive learning (typical in neural networks). DE (Differential Evolution) is a very simple population-based stochastic function minimizer used in a wide range of optimization problems, including multi-objective optimization. SA (Simulated Annealing) is a classic trajectory descent optimization technique. Finally, CHC is a particular class of evolutionary algorithm which does not use mutation and relies instead on incest prevention and disruptive crossover. Due to the complexity of such a large analysis including so many techniques, we have used not only sequential algorithms, but also grid computing with BOINC in order to execute thousands of experiments in only several days using around 100 computers.

Keywords: Omnidirectional BTS, RND, PBIL, DE, SA, CHC.

1 Introduction

The Radio Network Design problem is a kind of telecommunication network design problem. When a set of geographically-dispersed terminals needs to be covered by transmission antennas (also called base station transmitters or base transceiver stations -BTS-), a capital subject is to minimize the number and locations of those antennas while covering the largest possible area.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 853 – 860, 2007.
© Springer-Verlag Berlin Heidelberg 2007

RND is an NP-hard problem; therefore its solution by means of evolutionary algorithms is appropriate. In this work we use several different evolutionary approaches in order to solve this problem: PBIL, DE, SA and CHC.

Finally, since our interest is not only studying these techniques, but also to open new research lines, we have applied grid computing for the realization of our experiments (not in an exhaustive manner for space constraints in this paper). In particular, we have used BOINC, a very interesting proposal for volunteer computing and desktop grid computing.

The rest of the paper is organized as follows: Section 2 briefly explains the RND problem with omnidirectional BTS. After that, in the following section we introduce the PBIL, DE, SA and CHC algorithms. Then, in section 4 we show the most interesting results of this work, including comparisons among the different studied techniques, finally leading to the conclusions and future work in the last section.

2 Radio Network Design with Omnidirectional BTS

The RND problem [1,2] consists in covering the largest area with a minimal set of transmitters. In order to mathematically define this problem, let us consider the set L of all potentially covered locations and the set M of all potential transmitter locations. Let G be the graph, (M $\cup$ L, E), where E is a set of edges such that each transmitter location is linked to the locations it covers. As the geographical area needs to be discretized, the potentially covered locations are taken from a grid, as shown in figure 1a. In our case, we focus on a 287×287 point grid representing an open-air flat area and we will be able to use a maximum of 349 available locations for placing antennas.

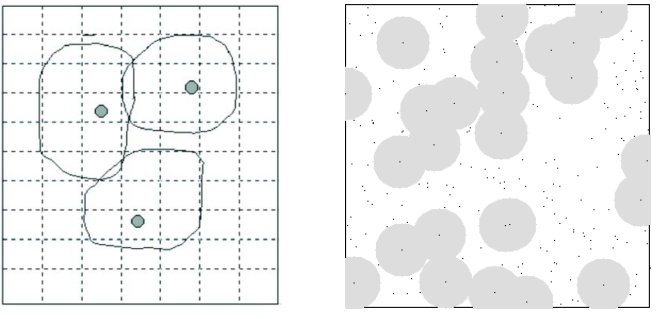


Fig. 1. (a) Associated covered cells of 3 potential transmitters. (b) Omnidirectional BTSs.

The goal of RND is to search for the minimum subset of transmitters that covers a maximum surface of an area, therefore, we are searching for a subset M' $\subseteq$ M such that |M'| is minimum and such that |Neighbours(M', E)| is maximum, where:

$$\text{Neighbours}(M', E) = \{u \in L \mid \exists v \in M', (u, v) \in E\} \quad (1)$$

To achieve this, we use the fitness function shown in equation 2 [3].

$$f(x) = \frac{CoverRate(x)^2}{NumberTransmittersUsed(x)} \quad (2)$$

An important constraint in this problem consists in determining the list of available locations for the antennas, because there are some places where the antennas can not be placed (public recreation areas, etc.). In our case, for the predefined set of available locations we selected the one included in [4]. This will make easy the comparisons among the different evolutionary techniques.

In our experiments we consider omnidirectional BTS (see figure 1b) and each transmitter has an associated coverage of a 22-sector-radius circle.

3 Different Evolutionary Approaches

In this section we briefly introduce the different evolutionary algorithms used for solving the RND problem.

3.1 PBIL

Population-Based Incremental Learning (PBIL) is a method that combines a genetic algorithm with competitive learning for function optimization. Instead of applying operators, PBIL infers a probability distribution from the present population and samples the new population from the inferred distribution [5,6].

3.2 DE

Differential Evolution (DE) is an algorithm that targets continuous optimization problems and has been used in the past with satisfactory results [7,8]. DE is a simple population-based stochastic function minimizer/maximizer, used in a wide range of optimization problems, including multi-objective optimization [9]. It has been modified in this research to work with discrete representations [10].

3.3 SA

Simulated annealing (SA) is a generic probabilistic meta-algorithm for the global optimization problem, namely locating a good approximation to the global optimum of a given function in a large search space. It was independently invented by S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi in 1983 [11], and by V. Černý in 1985 [12]. SA is a trajectory based optimization technique (i.e., only one tentative solution is manipulated in contrast with the rest of algorithms here, where a population of solutions is used). It is commonly found in industry and provides good results; therefore it constitutes an interesting method for comparison.

3.4 CHC

The fourth algorithm we propose for solving the RND problem is Eshelman's CHC [13], which stands for Cross-generational elitist selection, Heterogeneous recombination (by incest prevention), and Cataclysmic mutation. CHC is a kind of

Evolutionary Algorithm (EA), where mutation is not used. As a mechanism for preventing convergence and maintaining diversity, CHC employs incest prevention [14] plus a special recombination procedure known as HUX, combined with a special restarting mechanism for adding diversity through mutation when stagnation is detected (cataclysmic mutation).

4 Results

In this section we present the most interesting results coming from using each of the evolutionary techniques we have proposed to solve the RND problem.

4.1 Results with Population-Based Incremental Learning (PBIL)

In this work we have first evaluated different configuration parameters for our PBIL algorithm in order to solve the RND problem since this application to this problem is relatively new. In particular, the parameters we can adjust in PBIL are: number of samples (the population size), mutation probability, mutation shift (intensity of the mutation that affects the probability vector), learning rate, and whether an elitist strategy is used or not (the best individual in the previous population is transferred to the current generation unaltered). As we can see, the total number of possible combinations is high, for this reason we have used the middleware system BOINC [15,16] (Berkeley Open Infrastructure for Network Computing) in order to perform massive computations/experiments in a parallel way for PBIL. In this way, we can do a deep survey about which are the best parameter values for solving the RND problem with PBIL, which is needed to guide future research after this first study. Researchers interested in learning more about our platform RND-BOINC (RND@home), or wanting to join in this project (volunteer donation of CPU cycles), can access it via the website http://arcoboinc.unex.es/rnd. At present, around 100 computers are available in this project, executing hundreds of experiments at the same time.

Table 1 shows the most important results using PBIL. As we can see, PBIL obtains a reasonable result (85% of coverage with only 62 transmitters) with a low computational effort (only 333,045 evaluations).

Table 1. Results with PBIL for solving the RND problem (omnidirectional BTSs)

	Config. best result
# Generations	2500
# Individuals	135
Mutation probability	0.02
Mutation shift	0.05
Learning rate	0.10
Elitism	NO

	Best result
Fitness function	116.95
# Transmitters	62
Coverage	85%
Execution time	2 h, 31', 9''
Execution on	Pentium IV – 2.8 GHz
# Evaluations	333,045

4.2 Results with Differential Evolution (DE)

In order to compare the results we have obtained from DE with other evolutionary approaches, it is necessary to apply the experiments on the same predefined set of

BTS available locations. In the case of DE, the parameters we can adjust are: number of generations, population size, and the crossover function to use. Two crossover functions have been considered: FA and SA. Let be two set of locations (individuals), named A and B (the parents). Let be the S individual (the offspring) obtained from the application of the crossover function to A and B. FA function chooses the first half of A to build the first half of the offspring. The second half of the offspring is then built with the first half of B, but if a repeated location appears, successive locations of the second halves of B and A are taken. SA function chooses the second half of A to build the first half of the offspring, and the second half of the offspring is then built with the second half of B, but if a repeated location appears, successive locations of the first halves of B and A are taken.

Table 2 shows the most important results, considering the same instance of the problem used by the other evolutionary algorithms presented in this paper. The main conclusion is that the desired optimal coverage (100%) has not been reached. Also, it can be observed that DE algorithm is very fast.

Table 2. Results with DE for solving the RND problem (omnidirectional BTSs)

	Config. best result
# Generations	4000
# Individuals	2000
Crossover	FA

	Best result
Fitness function	100.54
# Transmitters	52
Coverage	72.30%
Execution time	5', 33''
Execution on	Pentium IV –1.7 GHz
# Evaluations	4,914

4.3 Results with Simulated Annealing (SA)

SA has been used on the same instance as the previous algorithms, in order for the obtained results to be comparable.

SA has only three parameters the programmer needs to tune (we use 1 as initial temperature): mutation probability, length of the Markov chain, temperature decay (α). The length of the Markov chain and the temperature decay have been proven to work in the same manner, thus to be equivalent. Therefore, we decided to keep the first at a constant value of 50, and allow the tuning of the latter.

Table 3 shows the results. The tests have been performed in a 16 machine cluster named in dedicated mode, and the code has been developed using the MALLBA library [17]. This resource code is available at the web page http://neo.lcc.uma.es/mallba/easy-mallba/index.html.

Table 3. Results with SA for solving the RND problem (omnidirectional BTSs)

	Config. best result
# Evaluations	50,000,000
Mutation probability	0.005
Markov chain length	50
Temperature decay	0.99998
Initial temperature	1

	Best result
Fitness function	157.77
# Transmitters	52
Coverage	90%
Execution time	2 h, 16', 20''
Execution on	Pentium IV – 2.4 GHz
# Evaluations	4,152,235

SA has been able to solve the problem with 90% coverage (the best result until now), but presents a high computational effort with 4,152,235 evaluations.

4.4 Results with CHC

When using the CHC algorithm on the same instance that the previous methods, we have considered two parameters that can be tuned (the rest of configuration parameters are fixed to their typical values): population size and cataclysmic mutation probability. Table 4 shows the best configuration, and the results obtained. The tests have been performed in a 16 machine cluster named in dedicated mode, and the code has been developed using the MALLBA library [17]. This resource code is available at the web page http://neo.lcc.uma.es/mallba/easy-mallba/index.html.

During the tests we have concluded that the mutation probability (second parameter tuned) has little effect on the algorithm's performance, and can be kept at a value of 35% without any significant loss of efficiency.

CHC solved the problem with 90% coverage (a very good result), but it presents a quite high computational effort (though lower than SA).

Table 4. Results with CHC for solving the RND problem (omnidirectional BTSs)

	Config. best result
# Evaluations	50,000,000
# Individuals	8,000
Mutation probability	0.35
Incest distance	25% vector length
Crossover probability	0.8
Elitism	YES

	Best result
Fitness function	157.77
# Transmitters	52
Coverage	90%
Execution time	31', 17''
Execution on	Pentium IV – 2.4 GHz
# Evaluations	1,400,000

5 Conclusions and Future Work

In this paper we have solved the RND (Radio Network Design) problem with different evolutionary approaches. In particular, we have focused on the RND using omnidirectional BTS. Our aim was to solve the problem efficiently and at the same time research in the results of a wide spectrum of modern techniques. All these evolutionary algorithms have obtained reasonable results (from 72.30% to 90% of coverage). However, there are important differences.

If we look for the best fitness value (relation between coverage and number of antennas), the best alternatives are CHC and SA (see figure 2). Between them, CHC has proven to be better suited than SA to solve the problem instances, since it obtains an equivalent result, but the computational effort (#evaluations) is smaller. On the other hand, DE obtains the worst fitness value. However, DE is the evolutionary approach that needs the lowest number of evaluations (computational effort) in order to obtain a reasonable result. The difficulties in DE's accuracy may be originated by its intrinsic continuous nature, originating the stagnation of its progression [18].

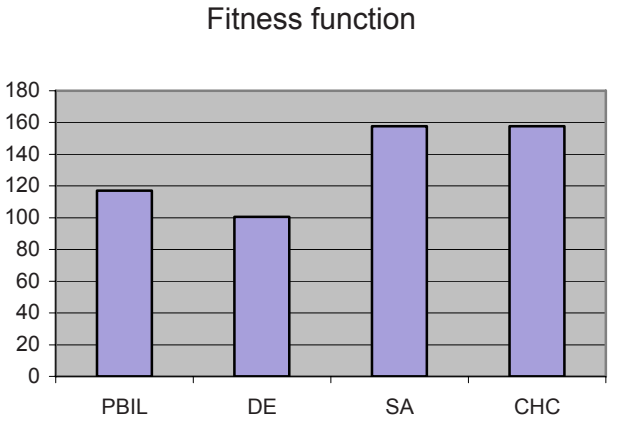


Fig. 2. Fitness value for each evolutionary approach

Future work includes the study of more sophisticated evolutionary algorithms by communicating information among component parallel agents. Also, more complex and realistic problem instances will be tackled using the best performing techniques. The study of a real-size instance of the coverage of a city is in the current agenda.

Acknowledgments. This work has been partially funded by the Spanish Ministry of Education and Science under contract TIN2005-08818-C04-01 and TIN2005-08818-C04-03. Guillermo Molina is supported by grant AP2005-0914 from the Spanish government.

References

1. Calégari, P., Guidec, F., Kuonen, P., Kobler, D.: Parallel Island-Based Genetic Algorithm for Radio Network Design. Journal of Parallel and Distributed Computing 47(1), 86–90 (1997)
2. Calégari, P., Guidec, F., Kuonen, P., Nielsen, F.: Combinatorial Optimization Algorithms for Radio Network Planning. Theoretical Computer Science 263(1), 235–265 (2001)
3. Alba, E.: Evolutionary Algorithms for Optimal Placement of Antennae in Radio Network Design. NIDISC 2004 Sixth International Workshop on Nature Inspired Distributed Computing, IEEE IPDPS, Santa Fe, USA, pp. 168–175 (April 2004)
4. OPLINK: (May 2007), http://oplink.lcc.uma.es/problems/rnd.html
5. Baluja, S.: Population-based Incremental Learning: A Method for Integrating Genetic Search based Function Optimization and Competitive Learning. Technical Report CMU-CS-94-163, Carnegie Mellon University (June 1994)
6. Baluja, S., Caruana, R.: Removing the Genetics from the Standard Genetic Algorithm. 12th Int. Conference on Machine Learning, San Mateo, CA, USA, pp. 38–46 (May 1995)
7. Price, K., Storn, R.: Differential Evolution – A Simple Evolution Strategy for Fast Optimization. Dr. Dobb's Journal 22(4), 18–24 (1997)
8. Price, K., Storn, R.: DE website (May 2007), http://www.ICSI.Berkeley.edu/~storn/code.html

9. Abbass, H.A., Sarker, R.: The Pareto Differential Evolution Algorithm. Int. Journal on Artificial Intelligence Tools 11(4), 531–552 (2002)
10. Mendes, S., Gómez, J.A., Vega, M.A., Sánchez, J.M.: The Optimal Number and Locations of Base Station Transmitters in a Radio Network. In: 3rd Int. Workshop on Mathematical Techniques and Problems in Telecommunications, Leiria, Portugal, pp.17–20 (September 2006)
11. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220(4598), 671–680 (1983)
12. Cerny, V.: A Thermodynamical Approach to the Travelling Salesman Problem: an Efficient Simulation Algorithm. Journal of Optimization Theory and Applications 45, 41–51 (1985)
13. Eshelman, L.J.: The CHC Adaptive Search Algorithm: How to Have Safe Search when Engaging in Nontraditional Genetic Recombination. Foundations of Genetic Algorithms, pp. 265–283. Morgan Kaufmann, San Francisco (1991)
14. Eshelman, L.J., Schaffer, J.D.: Preventing Premature Convergence in Genetic Algorithms by Preventing Incest. 4th Int. Conf. on Genetic Algorithms, CA, USA, pp. 115–122 (1991)
15. BOINC: (May 2007), http://boinc.berkeley.edu
16. Anderson, D.P.: BOINC: A System for Public-Resource Computing and Storage. 5th IEEE/ACM Int. Workshop on Grid Computing, Pittsburgh, USA, pp. 365–372 (November 2004)
17. Alba, E., Almeida, F., Blesa, M., Cotta, C., Díaz, M., Dorta, I., Gabarró, J., León, C., Luque, G., Petit, J., Rodríguez, C., Rojas, A., Xhafa, F.: Efficient Parallel LAN/WAN Algorithms for Optimization: The MALLBA Project. Parallel Computing 32(5-6), 415–440 (2006)
18. Lampinen, J., Zelinka, I.: On Stagnation of the Differential Evolution Algorithm. 6th International Mendel Conference on Soft Computing, MENDEL 2000, Brno, Czech Republic, pp. 76–83 (June 2000)

Fast Generation of Production Schedules on a Single Machine

Andreas Weidenhiller and Herbert Jodlbauer

FH-Steyr, Wehrgrabengasse 1-3,
A-4400 Steyr, Austria
{andreas.weidenhiller, herbert.jodlbauer}@fh-steyr.at
www.fh-steyr.at

Abstract. Many approaches to production scheduling are based on the step by step improvement of a given schedule. For these, finding a "good" initial schedule may critically influence performance. In the following, an extremely fast and simple method is presented for finding such a schedule in the case of a continuous-time single machine problem with the goal of minimizing setup and holding costs. In the case where there are no setups, this method always finds the optimal solution. A benchmark illustrates the possibilities of the method.

1 Introduction

Most problems in lot-sizing and scheduling are NP-hard, i.e. in the worst case the solution time may be exponential in the problem size. Well known problems of this kind are the discrete-time problems DLSP [10] and CLSP [3], and the continuous-time problem ELSP [6].

Because of the complexity of the problems, it is interesting to research polynomial time approximation algorithms (cf. [4, 12]) as well as more easily accessible special cases like single item problems (for a review see [1]). Although even among the single item problems many are NP-hard [1], there are some notable exceptions:

The continuous-time problems Economic Order Quantity [5] and Economic Production Lot [11] can both be solved explicitly. The seminal discrete-time contribution by Wagner and Whitin [13] can be solved in polynomial time.

In the paper at hand, it will be shown that for a continuous-time model (description in Section 2) under the simplifying assumptions of zero setups and discrete demand the globally optimal solution can be constructed in polynomial time (Section 3). Results on how the applied method performs in cases with nonzero setups are summarized in Section 4; directions for further research are indicated in Section 5.

2 Model

In this section, those aspects of the underlying model which are relevant to the article at hand are presented; for the full model see [7, 8].

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 861 – 869, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Figure 1a displays the cumulated demand $D_i(t)$ for a problem with $m = 2$ products within the time horizon $[0,T]$. $D_i(t)$ is monotonously increasing; left of 0 it assumes the value Zero, right of T the constant value $D_i(T)$. Clearly, Figure 1a shows a discrete demand structure; due dates correspond to discontinuities in the cumulated demand. Only a finite number of due dates are allowed.

For each product a constant production rate c_i is assumed, and so the cumulated production $X_i(t)$ for each product is piecewise linear, alternating between the rates zero and c_i (see Figure 1b). Time intervals where $X_i(t)$ is increasing with rate c_i are production intervals for the product in question. The amount produced in one such interval is called a *lot*. No backlog is allowed in the model, and so cumulated production must not fall below cumulated demand at any time *t*:

$$\forall i : \forall t \in [0,T] : X_i(t) \geq D_i(t) \ . \tag{1}$$

$X_i(t) - D_i(t)$ indicates the amount of product *i* which is in stock at time *t*.

Furthermore, it is clear that producing more of some product *i* than is required in total causes unnecessary costs. Thus it is no severe restriction to forbid excess production a priori:

$$\forall i : X_i(T) = D_i(T) \ . \tag{2}$$

Cumulated production without limitation of generality can be assumed to be zero at the beginning of the planning horizon:

$$\forall i : X_i(0) = 0 \ . \tag{3}$$

Immediately before the production of any lot, appropriate setup has to be performed, which for a changeover from product *i* to product *j* takes t_{ij}^S time units and costs k_{ij}^S monetary units. Only one product at any time can be dealt with, and for this product, there can be either setup or production activity, but not both.

A *schedule* (i.e. the information when to produce how much of which product) is unambiguously defined if the cumulated production functions for all products are given. If some schedule satisfies all of the conditions above, it is called *feasible*. Among all feasible schedules, the aim is to find an *optimal schedule*, i.e. one which minimizes the cumulated setup and holding costs. Setup costs can be calculated by simply adding them up, and the holding costs are obtained by multiplying the holding cost factor k_i^I with the area between X_i and D_i:

$$cost = \sum_{i,j} \sum_{\substack{\text{changeovers} \\ \text{from } i \text{ to } j}} k_{ij}^S + \sum_i k_i^I \int_0^T X_i(t) - D_i(t)\, dt \ . \tag{4}$$

In the paper at hand, only the holding costs will receive further attention.

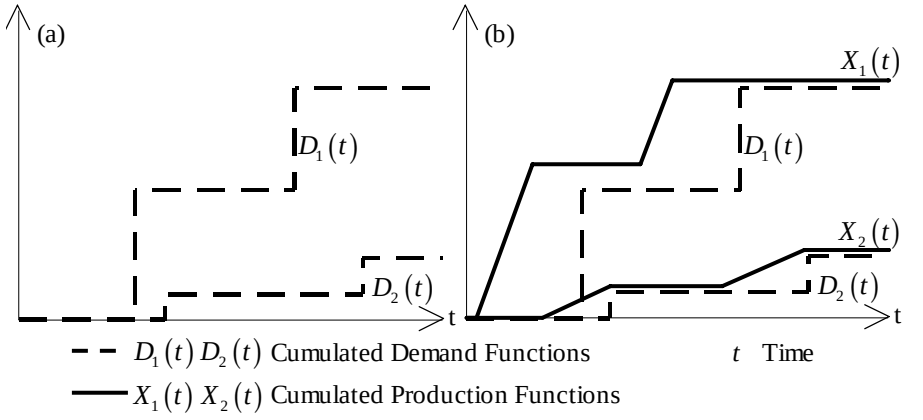


Fig. 1. Representation of demand and production in the model

3 An Optimal Schedule

In this section, a feasible schedule for the problem is created and proven to be globally optimal. The concept of 'priorities' used in this section enables a more concise formulation of the *Zero-Excess-Property* and should not be confused with the term 'priority rules' which is sometimes used as a synonym for dispatching rules.

Definition 1. Product *j* has *strictly higher priority* than product *i* if $c_j k_j^I > c_i k_i^I$. Furthermore, every product *i* with $c_i k_i^I > 0$ has a strictly higher priority than idle time.

Definition 2. A schedule is said to have the *Zero-Excess-Property*, if the following is true for the schedule: During the production of some product *i* as well as during idle time, no product with a strictly higher priority is in stock.

By requiring a schedule to have the Zero-Excess-Property, it is already determined (almost) unambiguously, as will be shown in the following by constructing such a schedule.

For the purpose of schedule construction and without limitation of generality, assume that the product types are numbered according to the following rule:

$$c_j k_j^I > c_i k_i^I \Rightarrow j > i \ . \tag{5}$$

The construction starts at the last due date (no production may occur after it) and moves backwards in time until the schedule is complete and all demand is satisfied.

Suppose the following situation occurs during schedule creation: At some time t_0, there is demand for two products *i* and *j*, with $c_j k_j^I > c_i k_i^I$. If product *i* is scheduled next before t_0, i.e. in some nonempty interval $[a, t_0]$, then *j* will have to be produced still earlier and kept in stock during $[a, t_0]$ in order to satisfy the demand at t_0. This is clearly a violation of the Zero-Excess-Property. Scheduling idle time during $[a, t_0]$

and producing *i* and *j* before *a* also would violate the Zero-Excess-Property; and so a decision on what to produce is required only if $c_j k_j^I = c_i k_i^I$ for two products *i* and *j*.

Consider the finite set $U = \{u_1 < u_2 < \ldots < u_J\}$ of all due dates which occur in the problem. Between two adjacent points of *U*, stock levels only change because of production. For constructing a schedule which satisfies the Zero-Excess-Property, the following method can thus be applied; it terminates after at most mJ iterations:

Algorithm 3.

1. Reassign the product indices *i* so that they satisfy (5).
2. Consider separately each interval $[u_{k-1}, u_k]$ between adjacent due dates, starting with $[u_{J-1}, u_J]$ and ending with $[0, u_1]$.
3. Within $[u_{k-1}, u_k]$ and beginning at u_k, schedule consecutive production intervals from the product *m* (highest priority) down to the product 1 (lowest priority), until either time u_{k-1} is reached or all demand from time u_k on is satisfied.
4. Demand which could not be satisfied during $[u_{k-1}, u_k]$ (i.e. demand of the products with the lowest priority values) will have to be satisfied in an earlier interval ($[u_{k-2}, u_{k-1}]$ or $[0, u_1]$) – accordingly, return to step 3 with $k \mapsto k-1$.
5. If there remains some unsatisfied demand even at time 0, there is no feasible schedule for the problem at all. The reason is that the algorithm satisfies all demand as late as possible, and that the time required for the production is independent of the sequence and number of lots because there are no setup times.

In the following proposition, it will be proven that the schedule $\overline{S}$ constructed by Algorithm 3 is cheaper or at least not more expensive than any other schedule *S*. The cumulated production functions defining $\overline{S}$ will be denoted by $\overline{X}_i(t)$.

Proposition 4. Consider a problem with zero setups and purely discrete demand, and a feasible schedule *S* for this problem. There exists a finite cost-decreasing sequence of schedules S^k with $S^0 = S$ and $S^K = \overline{S}$, where $\overline{S}$ is the schedule constructed by Algorithm 3. Thus, no schedule can be cheaper than $\overline{S}$.

Proof:

The basic approach to constructing the sequence of schedules is simple: Starting at the time horizon *T* and moving backwards in time, the differences between S^k and $\overline{S}$ are reduced step by step by manipulating some lots while retaining feasibility and not increasing the costs.

First, some auxiliary quantities are defined for each step *k*:

- τ^k: Earliest point in time after which S^k and $\overline{S}$ are identical, i.e. $\forall i \forall t > \tau^k : X_i(t) = \overline{X}_i(t)$.
- $]\eta^k, \tau^k[$: Longest open interval in which there are no due dates and no production start or end times of S^k.

- $]\bar{\eta}^k, \tau^k[$: Longest open interval in which there are no due dates and no production start or end times of $\bar{S}$.
- $]\alpha^k, \beta^k[$: Longest interval with the properties: $\alpha^k < \beta^k < \tau^k$, in S^k throughout $]\alpha^k, \beta^k[$ the same product *i* is produced as in $\bar{S}$ throughout $]\bar{\eta}^k, \tau^k[$, in S^k product *i* is not produced within $]\beta^k, \tau^k[$, there are no due dates in $]\alpha^k, \beta^k[$.
- λ^k : The length of the shortest of the above intervals, i.e. $\lambda^k := \min\{\tau^k - \bar{\eta}^k, \tau^k - \eta^k, \beta^k - \alpha^k\}$.
- μ^k : Earliest point in time with the properties: $\mu^k \le \alpha^k$, there is no due date within $]\mu^k, \beta^k[$, in S^k there is no idle time within $]\mu^k, \beta^k[$, in S^k production throughout $]\mu^k, \beta^k[$ occurs exactly in the sequence of products $j+1$, $j+2$, … $i-1$, i (where *j* is the product which is produced throughout $]\eta^k, \tau^k[$ in S^k ; some products may be missing in the sequence, but none may occur more than once).

Throughout the interval $]\lambda^k, \tau^k[$, both in S^k and $\bar{S}$ exactly one product is produced (if idle time is considered as product 0; otherwise case distinctions between idle time and production would have to be introduced; the proof however works for both cases analogously). Furthermore, S^k and $\bar{S}$ differ in $]\lambda^k, \tau^k[$ according to the definition of τ^k , i.e. in $\bar{S}$ product *i* is produced, and in S^k product *j*. $\bar{S}$ has been created by means of Algorithm 3; because of (5), this implies $j < i$. In $\bar{S}$, there is no idle time between τ^k and the next greater due date, and production between τ^k and the next greater due date occurs exactly in the sequence of products *i*, $i+1$, … $m-1$, *m* (some products may be missing in the sequence, but none may occur more than once).

The schedule S^{k+1} is created from S^k in the following manner:

- S^{k+1} is identical to S^k throughout $]\tau^k, T]$, $]\beta^k, \tau^k - \lambda^k[$ and $[0, \mu^k[$.
- Throughout $]\tau^k - \lambda^k, \tau^k[$, product *i* is produced instead of product *j*.
- Previous production of product *i* is reduced accordingly: Product *i* is produced only throughout $]\alpha^k + \lambda^k, \beta^k[$ instead of throughout $]\alpha^k, \beta^k[$. No backlog can occur because of this, because no backlog occurs in $\bar{S}$. Furthermore, this decreases the holding costs for product *i* by $\text{cost}_1^- := c_i k_i^I \lambda^k \left(\tau^k - \lambda^k - \alpha^k\right)$.
- All production which in S^k occurs within $]\mu^k, \alpha^k[$ is shifted to the right by λ^k , i.e. in S^{k+1} it occurs within $]\mu^k + \lambda^k, \alpha^k + \lambda^k[$. No backlog can occur because of this: there are no due dates within $]\mu^k, \beta^k[$. Only products j' with $j' > j$ are

produced throughout $\left]\mu^k+\lambda^k,\alpha^k+\lambda^k\right[$; thus a lower limit for the achieved decrease (cost_2^-) in holding costs is given by $\text{cost}_2^- \geq c_j k_j^{\mathrm{I}} \lambda^k \left(\alpha^k-\mu^k\right)$.

- Instead of throughout $\left]\tau^k-\lambda^k,\tau^k\right[$, product j is produced throughout $\left]\mu^k,\mu^k+\lambda^k\right[$, i.e. much earlier. Thus no backlog can occur because of this, but the holding costs for product *j* are increased by $\text{cost}^+ := c_j k_j^{\mathrm{I}} \lambda^k \left(\tau^k-\lambda^k-\mu^k\right)$.

Because of (5), the total increase in costs from S^k to S^{k+1} can be estimated by

$$\Delta\,\text{cost} = \text{cost}^+ - \text{cost}_1^- - \text{cost}_2^- \leq \left(c_j k_j^I - c_i k_i^I\right)\lambda^k\left(\tau^k-\lambda^k-\alpha^k\right) \leq 0\ . \tag{6}$$

Thus S^{k+1} is cheaper or at least not more expensive than S^k; furthermore, S^{k+1} and $\overline{S}$ are identical after time $\tau^k-\lambda^k$. For the step from S^k to S^{k+1} care has been taken to maintain as closely as possible a sequence of production as follows: “idle time, product 1, 2, 3, … *m*, idle time, etc.”.

In order to be able to determine an upper bound *K* for the number of steps from S^k to S^{k+1}, imagine the interval $\left[0,\max U\right]$ (where *U* is the finite set of all due dates for all products) as partitioned into time slots. Each of them is reserved for the production of one predetermined product (or idle time) and they are arranged in the sequence “idle time, product 1, 2, 3, … *m*, idle time, etc.”. Slots may be empty (i.e. have zero length); they are spread over the interval so that due dates occur only at the borders between a slot for product *m* and the succeeding slot for idle time. These borders may not change from one step *k* to the next; all other borders between slots may change, but always preserving the sequence, and not increasing the number of slots. Following these rules, the number *N* of slots required to accommodate all schedules S^k which turn up in the sequence from *S* to $\overline{S}$ is bounded above by

$$N \leq (2J+n)(m+1)\ , \tag{7}$$

where *J* is the cardinality of *U*. Due to the choice of λ^k, in every step *k* at least one of the following occurs:

- $\tau^{k+1} \leq \eta^k$, i.e. in step $k+1$ an earlier time slot will be examined concerning the similarity between S^{k+1} and $\overline{S}$ (can happen at most in *N* steps).
- $\tau^{k+1} \leq \overline{\eta}^k$, i.e. in step $k+1$ an earlier lot of $\overline{S}$ (or, if there is a due date at $\overline{\eta}^k$, the same lot but left of this due date) will be examined concerning the similarity between S^{k+1} and $\overline{S}$ (at most in $\overline{n}+J$ steps, where $\overline{n}$ is the number of lots in $\overline{S}$).
- $\beta^{k+1} \leq \alpha^k+\lambda^k$, i.e. when next a quantity of product *i* has to be shifted to the right, an earlier time slot will have to be used (at most in Nm steps).

The total number of steps therefore is bounded above by $K := N(m+1)+\overline{n}+J$; the transformation from *S* to $\overline{S}$ is a finite process; and because the costs never increase from one step to the next, no cheaper schedule than $\overline{S}$ can exist. □

4 Benchmark Results

In this section, two benchmarks are presented in order to determine the degree to which Algorithm 3 still performs well when setup activities are no longer negligible.

As a measure of the holding costs' impact on total costs, the ratio of minimum holding costs to expected setup costs will be used. Expected setup costs are calculated as the average setup cost for a product changeover multiplied by the total number of non-zero order quantities. Minimum holding costs are calculated by considering each due date for each product separately, ignoring possible overlaps. Each due date for a product *i* corresponds to a step in the cumulated demand function, the height of which step equals the order quantity *v*. Minimum holding costs are achieved by starting production v/c_i time units before the due date, because then production will finish just in time. The area between cumulated production and cumulated demand in this case is a triangle of width v/c_i and height *v* and corresponds to $k_i^1 v^2/2c_i$ monetary units in holding costs.

In the first part, a comparison with the benchmark in Jodlbauer et al [8] is made. In this article, an optimization method of higher complexity and longer runtimes than Algorithm 3 has been tested. The benchmark consisted of nine instances with two to ten products and randomly generated parameters. The ratios of minimum holding costs to setup costs were very high, ranging from $2{:}10^{-5}$ to $2{:}10^{-3}$. It is therefore to be expected that the results of Proposition 4 apply very well to these problems. The results as displayed in Table 1 confirm this: The costs for the schedules produced by Algorithm 3 never are more than 0.4% above those of the cheapest schedule known for the respective problem.

Table 1. Performance of Algorithm 3 on the problems in Jodlbauer et al [8], in comparison to the best results known

Number of Products	2 to 4	5	6 to 8	9	10
Above best known by	0.00%	0.04%	0.07%	0.34%	0.09%

The second benchmark is based on 270 problems from Cattrysse et al [2] with 2 to 6 products and a planning horizon of 60 time units (or 'periods', in the discrete view). The performance of Algorithm 3 on this set of problems was compared to that of the algorithm devised by Jordan and Drexl [9] and is predictably poor due to the comparatively low ratios of minimum holding costs to setup costs: They range from 1:4 to 1:120 with an average of 1:21 (±70%). Moreover, Table 2 shows that in the presence of significant setup times, the Zero-Excess algorithm produces infeasible solutions for

Table 2. Performance of Algorithm 3 on the problems in Cattrysse et al [2], in comparison to the performance of the algorithm by Jordan and Drexl [9]

Total of problems	neither solves	Only [9] solve	Both solve	Objective of Zero Excess above [9] by (average)
270	20%	23%	57%	82%

43% of the problems, while the algorithm by [9] does so for only 20% of the problems: Setup times do not influence sequencing and batching in Algorithm 3.

Even though this second result is very unsatisfactory if considered as a result on *final* schedules, the method of Algorithm 3 may remain interesting for quickly producing a *start solution* which is then further improved by some advanced method (see e.g. [8]).

5 Conclusion and Further Research

In the article at hand, an algorithm for a continuous-time lot sizing and scheduling problem has been presented and it has been proven that this algorithm produces globally optimal schedules in the case with zero setups and discrete demand. The algorithm has been benchmarked on one problem set with negligible setup costs and on one with substantial setup costs, and performed excellently in the first and poorly in the second benchmark as predicted.

For further research it will be interesting to extend the algorithm (and perhaps the proof of optimality) to problems with non-discrete demand or nonzero setup costs. Moreover, the case of nonzero setup times should be examined more closely in view of the feasibility problems which are likely to occur in this case.

References

1. Brahimi, N., Dauzere-Peres, S., Najid, N.M., Nordli, A.: Single item lot sizing problems. European Journal of Operational Research 168(1), 1–16 (2005)
2. Cattrysse, D., Salomon, M., Kuik, R., van Wassenhove, L.N.: A dual ascent and column generation heuristic for the discrete lotsizing and scheduling problem with setup-times. Management Science 39, 477–486 (1993)
3. Florian, M., Lenstra, J.K., Kan, A.H.G.R.: Deterministic Production Planning Algorithms and Complexity. Management Science 26(7), 669–679 (1980)
4. Gavish, B., Johnson, R.E.: A fully polynomial approximation scheme for single-product scheduling in a finite capacity facility. Operations Research 38(1), 70–83 (1990)
5. Harris, F.W.: How many parts to make at once, Factory: The Magazine of Management 10/2. 135-136, Reprint, Operations Research 38(6), 947–950 (1913)
6. Hsu, W.L.: On the general feasibility test of scheduling lot sizes for several products on one machine. Management Science 29(1), 93–105 (1983)
7. Jodlbauer, H.: An approach for integrated scheduling and lot-sizing. European Journal of Operational Research 172(2), 386–400 (2006)
8. Jodlbauer, H., Reitner, S., Weidenhiller, A.: Reformulation and Solution Approaches for an Integrated Scheduling Model. In: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganà, A., Mun, Y., Choo, H. (eds.) ICCSA 2006. LNCS, vol. 3984, pp. 88–97. Springer, Heidelberg (2006)
9. Jordan, C., Drexl, A.: Discrete Lotsizing and Scheduling by Batch Sequencing. Management Science 44(5), 698–713 (1998)
10. Salomon, M.: Deterministic lotsizing models for production planning. Springer, Berlin (1991)

11. Taft, E.W.: The most economical production lot (Formulas for exact and approximate evaluation – handling cost of jigs and interest charges of product manufactured included). The Iron Age 101, 1410–1412 (1918)
12. Van Hoesel, C.P.M., Wagelmans, A.P.M.: Fully polynomial approximation schemes for single-item capacitated economic lot-sizing problems. Mathematics of Operations Research 26(2), 339–357 (2001)
13. Wagner, H.M., Whitin, T.M.: Dynamic version of the economic lot size model. Management Science 5(1), 89–96 (1958)

Determining Orbital Elements of Extrasolar Planets by Evolution Strategies

Andreas M. Chwatal and Günther R. Raidl*

Vienna University of Technology
Vienna, Austria
{chwatal,raidl}@ads.tuwien.ac.at

Abstract. After the detection of the first extrasolar planet (exoplanet) more than one decade ago we currently know about more than 200 planets around other stars and there are about twenty multi-planet systems. Today's most commonly used technique for identifying such exoplanets is based on radial velocity measurements of the star. Due to the increasing time span and accuracy of the respective observations, the measured data samples will more and more contain indications of multiple planetary companions. Unfortunately, especially in multi-planet systems, the determination of these exoplanets' orbital elements is a rather complex and computationally expensive data analysis task. It is usually formulated as an optimization problem in which the orbital parameters yielding minimal residues w.r.t. the measured data are to be found. Very likely, improved algorithms for (approximately) solving this problem will enable the detection of more complex systems. In this paper we describe a specialized evolution strategy for approaching this problem.

1 Introduction

One of the most popular methods for planet discovery is based on radial velocity (RV) measurements of the central star, performed by high resolution spectroscopy. The *stellar wobble* induced by the gravitational influence of planetary companions manifests in Doppler shifts of the spectral absorption lines. A series of RV-measurements in principle makes it possible to determine most of the orbital elements of existing planets: the semi-major axis ("distance") a, the minimal planetary mass $\tilde{m}$ [1], the orbital eccentricity e, the argument of perigee ω, and the periastron time t_0. Only the right ascension of the ascending node Ω and the inclination i are impossible to derive by this method.

To compute the RV of a star resulting from a specific set of orbital elements, the Kepler equation $E - e \sin E = \frac{2\pi(t+t_0)}{T}$ needs to be solved numerically in advance. Using the equation $\tan\frac{v}{2} = \sqrt{\frac{1+e}{1-e}} \tan\frac{E}{2}$, the radial velocity of the planet is given by $\frac{dz}{dt} = K\big[\cos(v+\omega) + e\cos\omega\big]$, where $K = \frac{2\pi}{T}\frac{a\sin i}{\sqrt{1-e^2}}$.

* This work is supported by the European RTN ADONET under grant 504438.

[1] Minimal due to the unknown inclination i. Therefore $\tilde{m}$ is the actual parameter to be determined, but we denote this term by $\tilde{m}$ for simplicity.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 870–877, 2007.
© Springer-Verlag Berlin Heidelberg 2007

T denotes the revolution period of the planet and is according to Kepler's third law given by $T^2 = \frac{4\pi^2 a^3}{G\cdot(M+m)}$ (where G denotes the gravitational constant). The radial velocity of a star (with stellar mass M) originating from the considered planet is then given by $v_z^* = -\frac{1}{M} \cdot \frac{dz}{dt}$.

Now, a simple RV-model can be formed by summing up the terms resulting from the respective planets, but the discrepancies to a two-body system can be better approximated by the use of Jacobi-coordinates (as for instance explained in [1]). For outer planets this approach also takes into account the mass of the more interior planets. Thus the total RV of the star is given by $v_z^* = \sum_i \xi_i v_i$ where $\xi_i = \frac{m_i}{M+\sum_{j=0}^{i} m_j}$. By N denoting the total number of planets in a system, the determination of a fit to the observed radial velocities is basically a minimization problem in roughly $P = N \cdot 5$ parameters. Additional parameters arise from offsets in the data-samples (see sec. 3.1). Having RV-Measurements at K distinct points, the objective function is given by

$$\chi^2 := \sum_{k=0}^{K} \left(\frac{\triangle_k}{\sigma_k}\right)^2. \tag{1}$$

where $\triangle_k$ denotes the difference between the model and the observed radial velocity at time i. Sometimes the relative value $\chi^2_{\text{red.}} = \chi^2/\nu$, where $\nu = K-P-1$ (P denoting the total number of fitting-parameters) is also used.

1.1 Traditional Data Analysis

The traditional data analysis is largely based on the Fourier transform. From this one gets the frequency spectrum of the RV-signal, which is a good starting point for sinusoidal data fits which can later on be improved to Keplerians. In the case of multiple systems this method is typically applied iteratively. After determining the characteristics of one planet, its contribution to the RV data is subtracted. With this residual data, the next planet is considered. This approach is often referred to as *cleaning algorithm*, see [2]. Obviously, such an iterative strategy can be highly misleading. Furthermore, the Fourier-based analysis has disadvantages like the initial assumption of circular orbits. Consequently a simultaneous fitting of Keplerians is a desired and more promising alternative.

2 Previous Work

The first application of Genetic Algorithms (GAs) in the area of exoplanet research has been mentioned in [3], where GAs with classical binary encodings are used to search for the neighborhood of a local minimum. In [4] a GA routine is used to verify the uniqueness of the 3-planet-solution to the υ-And System. In [5] the authors use GAs in combination with the Levenberg-Marquardt algorithm to drive numerical integrations to analyse a strongly interacting system. In [1] and [6] the authors describe the use of a penalty for unstable configurations. They use MEGNO (see [7]) for the fast indication of possible instabilities.

The authors mention, that *"[..]GAs are still not very popular but they have been proven to be very useful for finding good starting points for the precise gradient methods of minimization like e.g. the well known Levenberg-Marquardt scheme"* [1]. To summarize the hitherto efforts, standard GAs with binary encodings have been applied to the problem, and they have been augmented by gradient-based local methods and penalties, but no more problem-adequate representation and variation operators have been studied so far. In fact, a binary representation of continuous parameters is nowadays usually not considered a meaningful encoding for most problems due to its weak locality [8]. In [9], another GA, called *Stakanof* method, is mentioned and applied to the μAra system with four planets. To our knowledge, however, no further details are yet published on this GA.

3 Evolution Strategy: ExOD-ES

As, for instance, pointed out in [10] or [8], many researchers consider *evolution strategies* (ES) to be more effective in solving continuous parameter optimization problems than genetic algorithms. This is mainly due to the weak locality of the GAs binary encoding as well as the absence of self-adaptation mechanisms to exploit the topological structure of the fitness-landscapes. In the following we present a problem-specific evolution strategy, essentially following [11].

The approach is based on a (μ, λ)-ES with self-adaptation of strategy parameters [12] where μ denotes the size of the population and λ the number of offsprings created in each generation.

In evolution strategies mutation is considered the primary operator and selection is traditionally done in a deterministic way by choosing the μ best offsprings. Mutation (eq. 2) is performed by adding a Gaussian-distributed random number to each parameter, where the standard deviation is given by a strategy parameter σ_i, associated with each parameter. These strategy parameters are optimized themselves by also undergoing mutation. They are modified by the multiplication with a log-normally distributed random value (eq. 3).

$$x_i' = x_i + N_i(0, \sigma_i') \tag{2}$$

$$\sigma_i' = \sigma_i \cdot e^{N(0,\tau_0)+N_i(0,\tau)} \tag{3}$$

$N(0, \tau_0)$ is a normally-distributed random value that is calculated only once for each candidate solution, while $N_i(0, \tau)$ is sampled individually for each parameter. The corresponding standard deviations are less critical and chosen as usual: $\tau_0 \propto 1/\sqrt{2\sqrt{n}}$. For a more detailed description of these standard elements of evolution strategies, the reader is referred to e.g. [8], or [13]. In the following we describe the special properties of the ES for the determination of the orbital elements of exoplanets, subsequently referred to as ExOD-ES[2]

[2] ExOD-ES: Exoplanet Orbit Determination by Evolution Strategies.

3.1 Encoding

Each candidate solution represents a whole planetary system by orbital element vectors $p_i = (\tilde{m}_i, a_i, \omega_i, e_i, t_{0,i})$, $i = 1, \ldots, N$, where N is the pre-specified number of planets. As several data sets may contain offsets in their radial velocities, which may result from the movement of the star through space or instrument calibration, each sample involved in the fit introduces one additional parameter $rv_{\text{offset},d}$ $d = 1, \ldots, D$, with D denoting the total number of data samples. Thus we altogether have $P = N \cdot 5 + D$ parameters, and for each we additionally maintain a corresponding strategy parameter σ_d.

To avoid undesired symmetries in the representation, which degrade performance of the optimization substantially, we ensure that the orbital element vectors are always sorted according to ascending order of semi-major axes a_i.

3.2 Mutation

The classical (μ, λ)-mutation modifies all parameters of the offspring individuals before selection. In our case it turned out to be better to merely mutate only one parameter-subset corresponding to one planet, which can be seen analogous to the evolution of subsystems in nature. In the following we omit the index i denoting the respective planets for simplicity.

For better convergence rates it is profitable to further consider some mutual dependencies. If ω has been changed by an angle ϕ we update t_0 by eq. 4 in order to keep the initial phase information, and thus not bias t_0 by the variation of another parameter:

$$t_0 = t_0 + T(a) \cdot \frac{\phi}{2\pi}. \tag{4}$$

For similar reasons we adjust t_0 after changes of a:

$$t_0 = t_0 + 1/2 \cdot (T(a) - T(a')). \tag{5}$$

This especially assists the development of long period planets as changes of a are distributed more uniformly over the time interval.

Restriction of the solution space: To improve the overall convergence properties of the algorithm we restrict the solution space in such a way that unstable systems, which cannot persist for more than a couple of centuries, will almost never be created by the variation operators.

Of course there remains the low probability of observing such systems that are currently undergoing some change, i.e. there are strong gravitational interactions between the planets and therefore the Keplerian model of the orbits which basically treats the system as a sum of two-body problems is not valid any more. In this case it is necessary to perform numerical integration. To determine meaningful starting values, however, Keplerian models are still useful. Consequently it is important to be able to adjust the impacts of the stability criterion and thus the amount of restriction.

Hill-Stability Criterion: A very simple stability-criterion dates back to G.W. Hill , and states that the toruses defined by the regions around planetary orbits up to distances from the orbital trajectory of Hill-radius

$$r_H = a\left(\frac{m}{3 \cdot M}\right)^{1/3} \tag{6}$$

of two planets should not overlap. More formally, for two planets i and $i+1$ this can be written as

$$|a_i - a_{i+1}| > \zeta\big[r_H(a_i, m_i) + r_H(a_{i+1}, m_{i+1})\big], \tag{7}$$

where ζ is a parameter that, for instance, guarantees stability for timescales as the age of the solar system ($\approx$4.5 billion years) when ranging from 11 to 13. The physical meaning of the Hill-radius can be interpreted as the border of the region where the gravitational influence of the planet clearly dominates the effects from the central star. If the toruses of the so called *Hill-spheres* of two planets overlap, it is just a question of time, when a first/next dynamical interaction will occur. ζ from eq. 7 is thus a parameter of ExOD-ES, forcing two distinct orbits to have a specific minimal distance. Moderate values like $\zeta = 3, \ldots, 5$ turned out to be sufficient for the algorithm.

Mutation Operator: The mutation operator is parameterized by the surrounding semi-major axes (in this context denoted by a^- and a^+) and the respective weighted Hill-radii ($\tilde{r}_H^- = \zeta \cdot r_H^-$ and $\tilde{r}_H^+ = \zeta \cdot r_H^+$). If there are no surrounding planets, the parameters $a_{\min}$ and $a_{\max}$, which can be derived from the input data, are used respectively. The mutation of the semi-major axes of one planet is performed as follows:

$$\eta = N(0, \sigma) \tag{8}$$

$$a' = \begin{cases} a + \eta & \text{if} \quad a + \tilde{r}_H + \eta < a^+ - \tilde{r}_H^+ \text{ and } a - \tilde{r}_H + \eta > a^- + \tilde{r}_H^- \\ a & \text{if} \quad a + \eta < a^+ - \tilde{r}_H^+ \text{ and } a + \eta > a^- + \tilde{r}_H^- \\ & \text{but} \quad a + \tilde{r}_H + \eta > a^+ - \tilde{r}_H^+ \text{ and } a - \tilde{r}_H + \eta < a^- + \tilde{r}_H^- \\ a + \eta + 2 \cdot \tilde{r}_H^+ & \text{if} \quad a + \eta > a^+ - \tilde{r}_H^+ \\ a + \eta - 2 \cdot \tilde{r}_H^- & \text{if} \quad a + \eta < a^- + \tilde{r}_H^- \end{cases} \tag{9}$$

In the first case there is no overlap of the Hill-toruses of the planet under consideration with the ones of the neighboring planets. In this case there is no difference to the conventional mutation. Otherwise there are different grades of violating the Hill-criterion. If the Hill-torus overlaps with the one of a neighbor, then we keep the initial value. The other two cases treat the situations of stronger violation of the criterion, i.e. where the planet itself would intrude one neighboring Hill-torus. Here the mutation operator enables to "tunnel" through the neighboring Hill-torus and therefore takes up another position (according to its distance).

This is a very important mechanism to escape local optima. Suppose the situation that the most planets are well determined, but on one position a planet, which is difficult to determine, is still missing in the model. The above mechanism enables a quick adoption of an individual to such a better model.

Unfortunately it is very likely that the objective value will decrease in such situations, even when the planet moved closer to the correct position. This is because the latter individual can be expected not to be well adapted to the new local minima. To counteract this shortcoming, we modify the conventional selection such that it supports the evolution of such new subsystems. Therefore we introduce a new parameter γ for each individual, which indicates the number of generations this "path of evolution" should survive. During the following iterations only the "new" planet is modified in order to make this individual competitive with the other ones of the population as soon as possible. After such a mutation we initially set $\gamma = \gamma_{\max}$, and then in each generational step the γ-parameter of the respective individuals is decremented by one (if $\gamma > 0$). The related modifications of the selection mechanism are described in sec. 3.4.

There is an additional mechanism that accomplishes the same purpose: the planet that is least important for the model is removed from it, and a new planet is created randomly on another position. Again we set a $\gamma > 0$ to enforce an evolution of this planet.

The mutation of the mass m requires an additional mechanism to guarantee valid solutions (eq. 10). If the mutation of the mass would violate the Hill-criterion we keep the maximal possible value and perform the conventional mutation otherwise.

$$m' = \begin{cases} m + \eta & \text{if} \quad a + \tilde{r}_H(m+\eta) < a^+ - \tilde{r}_H^+ \\ & \text{and } a - \tilde{r}_H(m+\eta) > a^- + \tilde{r}_H^- \\ \text{otherwise:} & \operatorname{argmax}_\iota(m+\iota), \iota \in [0,\mu], \text{ such that} \\ & \text{the above conditions are satisfied.} \end{cases} \tag{10}$$

3.3 Recombination

We apply different variants of recombination, which are performed before the mutation. For each individual we perform an intermediate recombination of all strategy-parameters, which is due to [12] an important prerequisite for a well working self-adaptation:

$$\sigma'_k = u_k \sigma_{a,k} + (1 - u_k)\sigma_{b_i,k} \tag{11}$$

Here u_k is a random variable in the domain $[0, 1]$. For some low percentage ($\approx 10\%$) of the population we additionally perform parameter intermediate recombination.

Furthermore we employ another variant, which is only performed for multiple systems. Hereby for each position one randomly selected planet from randomly

selected individuals is inherited. Although this *planetary recombination* is not very effective in the sense that it produces a lot of promising offsprings, it is often the crucial mechanism for finding the global optimum. This works, because the population occasionally consists of various systems where some but not all planets are already identified correctly and in addition maybe reside on a wrong position. By creating new systems by combining planets from existing ones it is very likely to perform one ore more accurate recombinations during the evolution process which finally yield the global optimum. It turned out to be best to perform this operation for about 10% of the individuals of the population and then put them directly into the next generation population (instead of the offspring population).

We found almost no significant improvement of the overall search quality by using the recombination of the strategy parameters and a marginal significance for the other ones. Nevertheless by using the *planetary recombination* the global optimum was often found earlier than without it. Although helpful in some situations, as far as our tests indicate recombination generally did not turn out to be a key factor for a successful search.

3.4 Selection

As already mentioned, the standard deterministic selection mechanism (see e.g. [8]) is modified to support more obtrusively altered individuals to some degree. For this purpose we reserve $\lfloor \mu\pi^\gamma \rfloor$ places of the population for individuals with $\gamma > 0$. Experiments indicated that values $\pi^\gamma \approx 3$ are well suited. Then, in a first step the best $\mu - \lfloor \mu/\pi^\gamma \rfloor$ offsprings are added to the next generation population. The remaining places are successively filled by adding the best $\lfloor \mu/\pi^\gamma \gamma_{\max} \rfloor$ individuals with $\gamma = i$, where $i = 1, \ldots, \gamma_{\max}$. In case of stagnation we re-include the best-so-far solution to the population.

4 Results

Parameter values of $\mu = 50$, $\lambda = 5000$ turned out to be adequate for a reliable search and high solution quality. The algorithm converges after some hundreds of iterations most of the time. Assuming about 150 measurements for a 3-planet system it takes about a couple of hours to find high quality solutions.

As currently no benchmark data-sets exist and previous publications are not focused on algorithmic aspects, it is difficult to compare our approach to others quantitatively. Tests with artificially created data-sets clearly demonstrate the ability to solve multiplanet-systems consisting of up to four planets.

As an example for the application to real data-sets we applied our algorithm to the υ-Andromedae and the 55-Cancri systems [4,14]. In most of our test runs, ExOD-ES was able to obtain the same or very similar configurations as published in these previous works.

5 Conclusions

In this article we described an evolution strategy which has been specifically tailored for effectively fitting Keplerians to RV data. Besides various mechanisms to create promising offsprings the use of the Hill-stability criterion for the mutation operator considerably reduces the size of the configuration space and therefore supports an effective search. We evaluated the proposed algorithm on real and artificially created data. Results indicate that the ExOD-ES appears to be a promising approach to solve complex systems with high accuracy. Our next step will be to apply our algorithm to the μ Ara system [9].

References

1. Goździewski, K., Konacki, M., Maciejewski, A.J.: Where is the Second Planet in the HD 160691 Planetary System? Astrophysical Journal 594, 1019–1032 (2003)
2. Charbonneau, P.: Genetic Algorithms in Astronomy and Astrophysics. Astrophysical Journal Supplement 101, 309 (1995)
3. Lazio, T., Cordes, J.: Genetic Algorithms: Searching for Planets Around Pulsars. In: ASP Conf. Ser. 72: Millisecond Pulsars. A Decade of Surprise, p. 391 (1995)
4. Butler, R.P., Marcy, G.W., Fischer, D.A., Brown, T.M., Contos, A.R., Korzennik, S.G., Nisenson, P., Noyes, R.W.: Evidence for Multiple Companions to υ Andromedae. Astrophysical Journal 526, 916–927 (1999)
5. Laughlin, G., Chambers, J.E.: Short-Term Dynamical Interactions among Extrasolar Planets. The Astrophysical Journal 551, L109–L113 (2001)
6. Goździewski, K., Konacki, M., Maciejewski, A.J.: Orbital Solutions to the HD 160691 (μ Arae) Doppler Signal. Astrophysical Journal 622, 1136–1148 (2005)
7. Cincotta, P.M., Simó, C.: Alternative Tools to Study Global and Local Dynamics-Application to Galactic Dynamics. In: Gurzadyan, V.G., Ruffini, R. (eds.) The Chaotic Universe, Proceedings of the Second ICRA Network Workshop. Advanced Series in Astrophysics and Cosmology, vol. 10, p. 247 (2000)
8. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York (1996)
9. Pepe, F., Correia, A.C.M., Mayor, M., Tamuz, O., Couetdic, J., Benz, W., Bertaux, J.L., Bouchy, F., Laskar, J., Lovis, C., Naef, D., Queloz, D., Santos, N.C., Sivan, J.P., Sosnowska, D., Udry, S.: The HARPS search for southern extra-solar planets. IX. μ Arae, a system with four planets. Astronomy and Astrophysics 462, 769–776 (2007)
10. Schöneburg, E., F., H., Feddersen, S.: Genetische Algorithmen und Evolutionsstrategien. Addison-Wesley, London, UK (1994)
11. Chwatal, A.: Bestimmung der Bahnelemente von extrasolaren Planeten aufgrund von Radialgeschwindigkeitsmessdaten mittels evolutionärer Algorithmen. Master's thesis, Vienna University of Technology, Vienna, Austria (2006)
12. Schwefel, H.P.: Numerical Optimization of Computer Models. Wiley, Chichester (1981)
13. Bäck, T., Fogel, D., Michalewicz, Z. (eds.): Handbook of Evolutionary Computation. IOP Publishing Ltd. (1997)
14. Marcy, G.W., Butler, R.P., Fischer, D.A., Laughlin, G., Vogt, S.S., Henry, G.W., Pourbaix, D.: A Planet at 5 AU around 55 Cancri. Astrophysical Journal 581, 1375–1388 (2002)

Training Multiclass Classifiers by Maximizing the Volume Under the ROC Surface

Stephan Dreiseitl[1,2]

[1] Dept. of Software Engineering
Upper Austria University of Applied Sciences
A-4232 Hagenberg, Austria
stephan.dreiseitl@fh-hagenberg.at
[2] Dept. of Biomedical Engineering
University of Health Sciences, Medical Informatics and Technology
A-6060 Hall, Austria

Abstract. Receiver operating characteristic (ROC) curves are a plot of a ranking classifier's true-positive rate versus its false-positive rate, as one varies the threshold between positive and negative classifications across the continuum. The area under the ROC curve offer a measure of the discriminatory power of machine learning algorithms that is independent of class distribution, via its equivalence to Mann-Whitney U-statistics. This measure has recently been extended to cover problems of discriminating three and more classes. In this case, the area under the curve generalizes to the volume under the ROC surface.

In this paper, we show how a multi-class classifier can be trained by directly maximizing the volume under the ROC surface. This is accomplished by first approximating the discrete U-statistic that is equivalent to the volume under the surface in a continuous manner, and then maximizing this approximation by gradient ascent.

Keywords: Multi-class ROC analysis, machine learning, discrimination analysis.

1 Introduction

The use of receiver operating characteristic (ROC) curves in the evaluation of diagnostic tests has a long history in medical informatics, but has only recently started to receive research interest in the field of machine learning and pattern classification.

For simplicity and to use established terminology, we will in the following refer to the two classes to be distinguished by a classifier as the "negatives" and the "positives", respectively. For classifier outputs on a continuous scale, a threshold is needed to categorize the classifier output as predicting either negative or positive class membership. A *true positive* classification is one where a member of the positive class is correctly assigned to this class; a *true negative* is the same for the negative class. True-positive and true-negative rates refer to

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 878–885, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the fraction of these correct classifications in the respective classes. In situations where there is no threshold that allows the classifier to correctly classify all positives and all negatives, there is a tradeoff between true-positive and true-negative rates. Considerations such as losses associated with false predictions influence the choice of threshold.

ROC curves are a plot of a classifier's true-positive rate versus its false-positive rate (1−true-negative rate), as one varies the threshold between positive and negative classifications across the continuum. ROC curves can thus be seen as tools for investigating the tradeoffs associated with various choices of thresholds. This is particularly important in medical applications, where asymmetric losses are commonplace. A perfect test is one for which there exists a threshold such that the test, using that threshold, correctly classifies all positives and all negatives. Such a test, therefore, has true-positive and true-negative rates of 100%; the ROC curve of a perfect test passes through the upper left corner of the unit square. For imperfect tests, the point on the curve closest to the upper left corner represents a thresholded test that strikes a good balance between maximizing true-positive and true-negative rates. Asymmetric losses mean that a different threshold may be preferred.

In machine learning and systems theory in general, one is mainly interested in another aspect of ROC curves: It can be shown that the area under the ROC curve is the probability $\theta = P(X < Y)$ that the model correctly assigns a lower output value to a randomly chosen negative case X than to a randomly chosen positive case Y [1]. The AUC value thus serves as an indicator of a model's discriminatory power. Using $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ to denote realizations of the random variables X and Y, respectively, and $\mathbb{1}[\cdot]$ to denote the logical indicator function, it is immediately obvious that

$$\hat{\theta}_2 = \frac{1}{m \cdot n} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbb{1}[x_i < y_j] \tag{1}$$

is an unbiased estimator of $P(X < Y)$, and thus also of the area under the curve. It is well-known from statistics that this estimator is equivalent to the Mann-Whitney U-statistic which, in turn, is equivalent to a Wilcoxon rank-sum statistic [2]. A classifier that rates all negatives lower than all positives therefore achieves the highest possible AUC value of 1, while a guessing classifier has an AUC value of 0.5.

The area under the curve is generally considered a better measure of classifier performance than accuracy, because it is invariant to class prevalences and monotonic transformations of classifier outputs [3,4]. It is therefore not surprising that a number of publications have recently begun to use AUC as the objective function in classifier training [5,6,7,8].

In this paper, we build upon our work on extending standard two-class ROC analysis to the multiclass case [9]. We show how multiclass ranking classifiers (in the functional form of a multilayer neural network) can be optimized by maximizing an objective function that is a direct generalization of AUC and the Wilcoxon-Mann-Whitney statistic to more than two classes. The remainder of

this paper is organized as follows: In Section 2, we show how ROC methodology can be extended to multi-class classifiers. The gradient ascent algorithm for directly maximizing this measure is described in Section 3. An example demonstrating the validity of this approach is presented in Section 4, and concluding remarks are given in Section 5.

2 A Discrimination Measure For Multi-class Classifiers

The main idea for three-class ROC analysis is to establish a three-dimensional "ROC surface" that is a generalization of traditional ROC curves. In this paper, however, we are more interested in the volume under the surface (VUS) and its role as a discrimination measure. In analogy to the two-class case, the volume under the surface can then be given an interpretation as a Wilcoxon-Mann-Whitney statistic. For this, we note that in general, a three-class classifier provides posterior class membership probabilities as outputs, i.e., a probability triple (p_1, p_2, p_3) with $p_1 + p_2 + p_3 = 1$. All classifier outputs are thus restricted to a triangular plane in three-space. In order to derive a discrimination measure for more than two classes, one needs a generalization of the two-class "correct ordering" notion $X < Y$ for negatives X and positives Y. With such a generalization, which we will introduce below, one can then extend Eq. (1) to cover the three-class case as

$$\hat{\theta}_3 = \frac{1}{m \cdot n \cdot l} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} \mathbb{1}[\mathrm{cr}(x_i, y_j, z_k)] \,. \tag{2}$$

Here, x_i, y_j and z_k denote, respectively, m, n and l classifier output triples for classes 1, 2 and 3. The objective function is thus the fraction of all possible combinations of class 1, 2 and 3 estimates for which $\mathrm{cr}(x_i, y_j, z_k)$ holds, i.e., which are correctly rated. The notion of "correct rating", which generalizes the notion of "correct ordering" for two classes, is defined as

$$\begin{aligned} \mathrm{cr}(x, y, z) :\Leftrightarrow\ & \left(d_{123}^{xyz} < d_{132}^{xyz}\right) \wedge \left(d_{123}^{xyz} < d_{213}^{xyz}\right) \wedge \left(d_{123}^{xyz} < d_{231}^{xyz}\right) \wedge \\ & \left(d_{123}^{xyz} < d_{312}^{xyz}\right) \wedge \left(d_{123}^{xyz} < d_{321}^{xyz}\right) . \end{aligned} \tag{3}$$

The symbol d_{123}^{xyz} denotes the sum of Euclidean distances between estimate triples and the corners representing their classes, i.e.,

$$d_{123}^{xyz} = \left\|(1,0,0) - x\right\| + \left\|(0,1,0) - y\right\| + \left\|(0,0,1) - z\right\| .$$

The other symbols in Eq. (3) denote sums of distances of the estimate triples to permutations of the corners of the triangle. In other words, three estimate triples are correctly rated if and only if the sum of distances of all estimate triples to their true class corners is shorter than the sum of distances to all permutations of corners. This concept is illustrated graphically in Fig. 1.

As a further analogy to the two-class case, it should be noted that the correct rating of three estimate triples depends only on their relative positioning, and

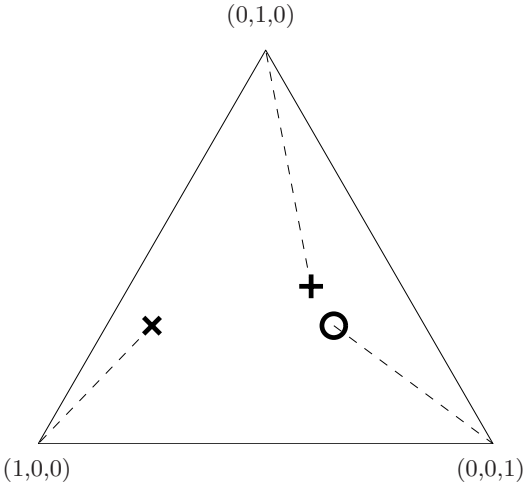


Fig. 1. Illustration of the concept of "correctly rating" three estimate triples $\times$, $+$ and $\bigcirc$ from classes 1, 2 and 3, respectively. A formal definition is given in Eq. (3).

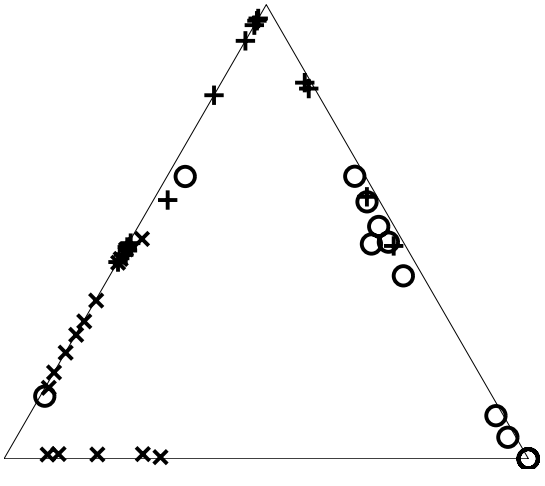


Fig. 2. Graphical representation of the outputs of a three-class neural network classifier, with $\times$, $+$ and $\bigcirc$ representing estimate triples from the three classes. The discrimination index is $\hat{\theta}_3 = 0.845$.

not on the absolute distances to their respective corners. This generalizes the notion of invariance to monotonic transformations of x_i and y_j values that is a characteristic of the two-class discrimination measure $\hat{\theta}_2$ in Eq. (1).

An example of a classifier trained to distinguish three classes from a synthetic data set is shown in Fig. 2.

3 Maximizing the Volume Under the Surface

Classifiers built to distinguish three or more classes are generally trained using a softmax error function. While the use of this error function allows the interpretation of classifier outputs as posterior class probabilities, it is not the same as directly maximizing a discriminatory measure to ensure that the separation between classifier outputs is as large as possible.

Maximizing the three-class discrimination measure $\hat{\theta}_3$ for an arbitrary underlying machine learning model amounts to solving a discrete optimization problem. The measure $\hat{\theta}_3$, as a sum of indicator functions, is not differentiable and therefore cannot be optimized using standard numerical methods. It is, however, possible to approximate $\hat{\theta}_3$ by replacing the discrete indicator function $\mathbb{1}[x < y]$ with a continuous approximation in the form of the logistic (sigmoid) function

$$s_\beta[x < y] = \frac{1}{1 + e^{-\beta(y-x)}}, \tag{4}$$

for given gain (steepness) parameter β. Even though this approximation, in theory, allows the calculation of the gradient of $\hat{\theta}_3$ with respect to the model parameters, the calculation is in practice complicated by the fact that $\hat{\theta}_3$ is not the sum of individual input pattern contributions, but a *combination* of these.

This complication can be overcome by noting that each of the five individual inequalities in the definition of in Eq. (3) has to be true for $\mathrm{cr}(x, y, z)$ to be true. This means that all the logistic representations have to approximate one, i.e., their sum has to exceed four. In detail, we can define the continuous approximation $\widetilde{\mathbb{1}}[\mathrm{cr}(x, y, z)]$ of $\mathbb{1}[\mathrm{cr}(x, y, z)]$ as

$$\widetilde{\mathbb{1}}[\mathrm{cr}(x, y, z)] = s_\beta\Big[4 < s_\beta\big[d_{123}^{xyz} < d_{132}^{xyz}\big] + s_\beta\big[d_{123}^{xyz} < d_{213}^{xyz}\big] + s_\beta\big[d_{123}^{xyz} < d_{213}^{xyz}\big] + s_\beta\big[d_{123}^{xyz} < d_{312}^{xyz}\big] + s_\beta\big[d_{123}^{xyz} < d_{321}^{xyz}\big]\Big],$$

and the continuous approximation $\tilde{\theta}_3$ to $\hat{\theta}_3$ as

$$\tilde{\theta}_3 = \frac{1}{m \cdot n \cdot l} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} \widetilde{\mathbb{1}}[\mathrm{cr}(x_i, y_j, z_k)] \,.$$

It is now possible to calculate the gradient of $\tilde{\theta}_3$ with respect to model parameters w as

$$\frac{\partial \tilde{\theta}_3}{\partial w} = \frac{1}{m \cdot n \cdot l} \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{l} \frac{\partial}{\partial w} \widetilde{\mathbb{1}}[\mathrm{cr}(x_i, y_j, z_k)] \,. \tag{5}$$

Although this derivation solves the optimization problem in theory (e.g., by a simple gradient-ascent algorithm), a practical implementation of this idea is hampered by two details: First, $\hat{\theta}_3$ and thus also $\tilde{\theta}_3$ depend on combinations of

classifier outputs, and not just on the individual outputs. This amounts to an exponential rise in the number of contributions to the sum in Eq. (5). In practice, this means that the applicability of this approximation is limited to small data sets. Second, the calculation of the gradient $\frac{\partial}{\partial w}\widetilde{\mathbb{1}}[\mathrm{cr}(x_i, y_j, z_k)]$ depends on the chosen model structure, and can be complicated by the nonlinear nature of the model. An example demonstrating the necessary gradient calculations is given in the next section.

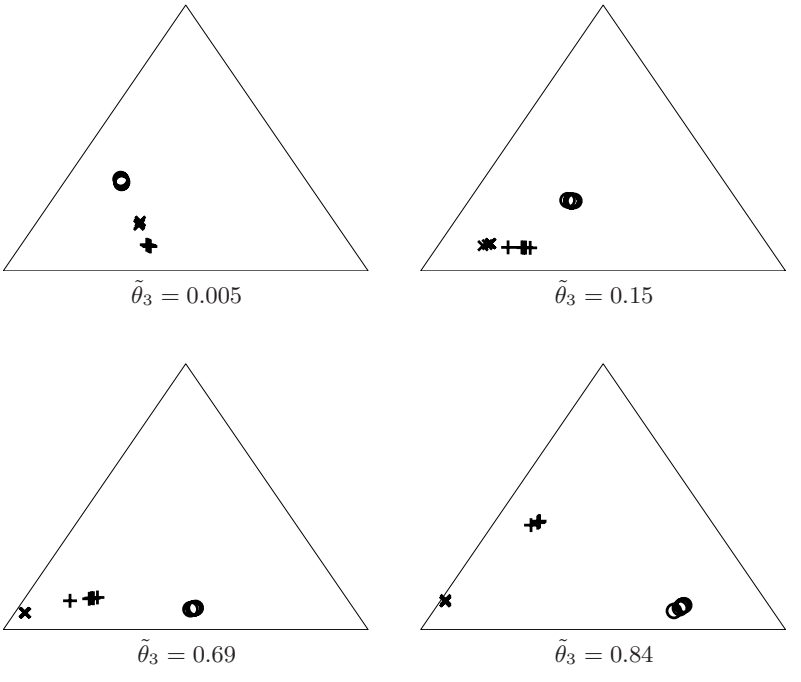


Fig. 3. Progress of classifier training by maximizing $\tilde{\theta}_3$. Shown are sets of estimate triples with their corresponding $\tilde{\theta}_3$ values at iteration $k = 10$ (*top left*), $k = 50$ (*top right*), $k = 70$ (*bottom left*), $k = 100$ (*bottom right*).

4 Example

In this section, we show how the approach presented above can be implemented to train a simple machine learning model to directly maximize the discrimination measure $\tilde{\theta}_3$. For the model, we chose a neural network with two input neurons, three hidden neurons, and three output neurons. The total number of parameters w in the model was thus 21. Starting with an arbitrary initial parameter setting w_0, we can use gradient ascent to maximize $\tilde{\theta}_3$ by iterating

$$w_{k+1} = w_k + \eta \frac{\partial \tilde{\theta}_3}{\partial w_k}$$

with appropriate learning rate η. Due to the nonlinear nature of the feedforward neural network, the gradient calculation cannot be performed by hand, but requires the use of a computer algebra system. With the system *Mathematica*, the gradient can be calculated automatically, but requires 10MB of storage. It is thus impractical to evaluate the gradient directly in *Mathematica*. Exporting the gradient calculation as C code, we implemented the gradient descent algorithm given above as a proof-of-concept using a data set of 15 two-dimensional data points (five for each class) and ran the gradient-ascent algorithm for 100 iteration steps with parameter settings $\beta = 2$ and $\eta = 1$. The resulting classifier outputs, together with their $\tilde{\theta}_3$ values, are shown in Fig. 3.

One can see that, as training progresses, the estimate triples move to their respective corners, with a corresponding increase in the value of $\tilde{\theta}_3$. It should be emphasized that this objective function maximizes discrimination between the three classes, and not calibration. This means that the estimate triples align in a configuration that discriminates well, but are not necessarily as close as possible to their corresponding corners. Maximizing discrimination *and* calibration could be achieved by adding a calibration term to the objective function. This, however, is outside the scope of this paper.

5 Conclusion

We introduced a measure $\hat{\theta}_3$ to assess the discriminatory power of three-class machine learning algorithms. We showed how this discrete measure, which is a variant of traditional Wilcoxon-Mann-Whitney statistics, can be approximated in a continuous manner. We presented a simple gradient ascent algorithm for maximizing the continuous approximation $\tilde{\theta}_3$ to $\hat{\theta}_3$ for an underlying neural network model, and demonstrated the validity of this approach on a toy problem.

The limitations of the method presented here are the exponential increase of computations required (in the size of the data set), which limits the applicability of this approach. Furthermore, the nonlinear nature of the continuous approximation $\tilde{\theta}_3$, together with the nonlinear nature of the underlying machine learning model, makes the gradient calculation cumbersome and lengthy.

Extensions of this work will deal with the introduction of second-order optimization methods on the basis of the gradient calculations presented here, and with sampling methods that could avoid having to use all combinations of estimate triples in the training process.

References

1. Bamber, D.: The area above the ordinal dominance graph and the area below the receiver operating characteristic graph. Journal of Mathematical Psychology 12, 387–415 (1975)
2. Hollander, M., Wolfe, D.: Nonparametric Statistical Methods, 2nd edn. Wiley-Interscience, USA (1999)

3. Ling, C., Huang, J., Zhang, H.: AUC: a statistically consistent and more discriminating measure than accuracy. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 519–526 (2003)
4. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proceedings of the 15th International Conference on Machine Learning, pp. 445–453 (1998)
5. Cortes, C., Mohri, M.: AUC optimization vs. error rate minimization. Advances in Neural Information Processing Systems 16, 313–320 (2004)
6. Rakotomamonjy, A.: Optimizing area under ROC curve with SVMs. In: Proceedings of the First International Workshop on ROC Analysis in Artificial Intelligence, pp. 71–80 (2004)
7. Herschtal, A., Raskutti, B.: Optimising area under the ROC curve using gradient descent. In: Proceedings of the 21th International Conference on Machine Learning, pp. 49–56 (2004)
8. Yan, L., Dodier, R., Mozer, M., Wolniewicz, R.: Optimizing classifier performance via the Wilcoxon-Mann-Whitney statistic. In: Proceedings of the 20th International Conference on Machine Learning, pp. 848–855 (2003)
9. Dreiseitl, S., Ohno-Machado, L., Binder, M.: Comparing three-class diagnostic tests by three-way ROC analysis. Medical Decision Making 20, 323–331 (2000)

Selective Association in Associative Memories – An Extension of the Model NEUNET

Reinhard Stumptner and Josef Küng

Johannes Kepler University of Linz
Institute for Applied Knowledge Processing (FAW)
Altenbergerstraße 69
4040 Linz, Austria
{rstumptner, jkueng}@faw.uni-linz.ac.at
http://www.faw.uni-linz.ac.at

Abstract. The neural-net-similar associative memory model NEUNET is able to store and reproduce patterns reading from its receptors. This article describes an enhancement of the model to associate faulty or incomplete inputs with weighted original patterns. The network is self-learning, the correction of inputs, the storage of new information and the calculation of weights which enables the model to store information with various priorities is self-organized. Priority of information is increased if it is frequently used and consequently in case of the association of unknown inputs it is more like to be associated with a high-priority original pattern. This selective behavior is the main enhancement of the model described in this contribution.

Keywords: Artificial intelligence, neural network, associative memory.

1 Introduction

The associative memory model NEUNET (neural network) [1] was developed in 1974 at the Johannes Kepler University in Linz by Professor Reichl. In consideration of the functionality of this model it can be termed a neural-net-similar associative memory system. This paper describes an enhancement of the model to associate inputs with weighted original patterns. The network is self-learning; the correction of inputs, the storage of new information and the calculation of weights which enables the model to store information with various priorities is unsupervised and self-organized. NEUNET consists of linked nodes/units, which can keep information and offer memory functionality. The connections between the nodes are bidirectional and not weighted. As an interface to environment the network has a certain number of receptors which are input and output units at the same time. Furthermore there exists a possibly big amount of hidden nodes which is extended during the process of learning new information. The original NEUNET models can store and reproduce inputs without losing information and associate faulty patterns with original ones [3]. Mostly these associations are ambiguous and in this case the association process chose the related original pattern in a not-deterministic way.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 886 – 893, 2007.
© Springer-Verlag Berlin Heidelberg 2007

So the idea of selective association is to replace this not-deterministic selection of associations by a deterministic method. As already mentioned the Selective NEUNET stores information in a weighted manner and consequently adds a property “priority” to information stored. The priority of information is increased if it is frequently used and when unknown inputs are processed it is more like to get an association with a high-priority original pattern. This enables the model to autonomously separate “important” information from a huge amount of any data.

2 Related Research

The field of neural networks is very wide spread. Mainly there are two different classes of problems to be solved with such models. On the one hand there is classification of data and on the other hand (auto-) association. Backpropagation [5] and the self-organizing maps (SOM) [2] of Kohonen are popular models mainly used for data classification or clustering. The components of these models are organized in multiple layers whereby each neuron gets the activation state of the lower level components as its input. Kohonen maps belong to the class of unsupervised learning systems. They are mainly used for finding clusters within known input data. For data association the most important model may be the Hopfield network [4]. Attractor networks used as associative memory models, like the Hopfield network, are concerning their behaviour the most similar ones to NEUNET. They can store patterns and associate new input with the most similar known pattern. Although the association process of NEUNET mostly is slower than that of the popular associative memory models like Hopfield, there are some main advantages like:

- The output of pattern completion and correction is nearer to the original patterns.
- The inversed pattern is not saved automatically.
- It is not necessary that patterns are orthogonal in pairs.

3 NEUNET-3

NEUNET-3 was the first enhancement of the original NEUNET model which had the ability to store and reproduce information without any losses. It is a self learning system, which means, that it has the ability to store new knowledge autonomously [1]. The units of the network have two properties called charge and stamp. The charge represents the current activation state of a neuron and can get one of the three values: +1: “on”; -1: “off”; 0: “unknown”. The stamp describes the experience of a node. It can get the values: -1: the unit never had a positive charge; and 0: the unit has had a positive charge. For linking nodes the model uses a special network unit, the aggregate, which will be described hereafter. If this network learns an unknown pattern, new nodes are generated and linked one with another. In addition to the receptor units the model has a set of very special, so called “counter-receptors”, whereby each receptor gets a counter-receptor always carrying the opposite charge of its corresponding receptor. The example in fig. 1 shows the basic behaviour of the NEUNET-3 model. A network with 5x5 receptors has been taught the patterns of the digits from

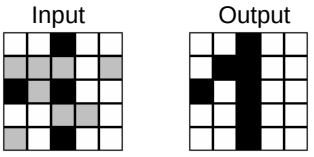


Fig. 1. Auto-association of an incomplete pattern. White pixels represent the charge "-1", black ones represent the charge "+1" and if a pixel is grey, this means that the charge is unknown ("0").

1 to 9. If the network now receives incomplete pieces of information, it tries to complete it according to known patterns.

As one can see in the figure above, the network is able to associate an incomplete input with the matching, known (original) pattern.

3.1 The Aggregate

As was mentioned before, the aggregate is a very special network-unit used for connecting nodes to each other.

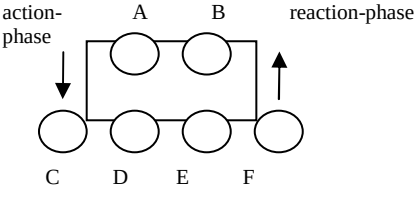


Fig. 2. Components of an aggregate

Table 1. Possible states of an aggregate

A	B	C	D	E	F
-1	-1	-1	-1	-1	+1
-1	0	-1	-1	0	0
-1	+1	-1	-1	+1	-1
0	-1	-1	0	-1	0
0	0	0	0	0	0
0	+1	0	-1	0	-1
+1	-1	-1	+1	-1	-1
+1	0	0	0	-1	-1
+1	+1	+1	-1	-1	-1

An aggregate consists of two input-nodes (A, B) and four output-nodes (C, D, E, F). It works in two phases, the action-phase and the reaction-phase, whereby charges are transported through an aggregate as described in table 1. The units are linked in a special way, so that the charges can be transported through the whole network. In the action-phase charges flow from the input-nodes to the output-nodes and in the following reaction-phase back to the input-nodes. Table 1 shows the input-output-behaviour of an aggregate.

3.2 Capabilities / Weaknesses of NEUNET-3

The model has the ability to store a theoretical unlimited number of different patterns autonomously, without losing any information. Incomplete inputs are completed if the association to the original pattern is clear. But what happens if this association is ambiguous, if more than one original pattern matches the (incomplete) input? The following example should demonstrate this problem (fig. 3 and fig. 4):

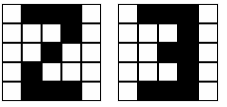


Fig. 3. Known patterns

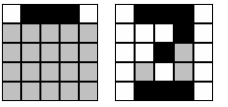


Fig. 4. Auto-association

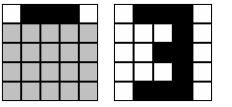


Fig. 5. Auto-association

The incomplete input of figure 4 matches “2” and “3”. The network now completes the input as far as possible, but pixels, where “2” and “3” have different values, keep their unknown charges (they stay grey). In this case the output pattern is a mixture, called the average of all matching original patterns. The problem of this situation is that drawing the conclusion from the output-pattern to the involved original patterns is not clear. Thus the model’s ability to complete patterns decreases the more different patterns the network knows. This may be one of the main weaknesses of the NEUNET-3 model. This problem the selective variant of NEUNET tries to overcome.

3.3 Selective Pattern Recognition

If an incomplete input-pattern matches more than one known pattern in the selective model, then the input should be associated to the one matching pattern, that’s known “best”, which is the basic idea of this model. A pattern is known better than another one, if it has been presented to the network more frequently. The selective NEUNET is able not only to store patterns reading from the receptors, additionally the patterns are stored weighted, whereby, whenever the network recognizes an input, the weights of the associated nodes are increased. The basic strategy of the selective NEUNET model is to strengthen information, which is “used” frequently and to forget information, which isn’t “reused” and so doesn’t seem to be important. The example in figure 5 shows the meaning of selective pattern recognition. The incomplete input matches both patterns, “2” and “3”, but the network knows “3” better than “2”, it has seen “3” more frequently, so it associates the input with “3”. This decision of the network is not fixed, if it would see “2” more often, then it would associate the same incomplete input with “2” later on.

4 The Probability-Based Approach

Remembering the behaviour of an aggregate one can see that the nodes C to F always react to special input-patterns (see table 1). An aggregate leaned three patterns, “++”, “+-” and “--” for instance, it knows all possible patterns except “-+”, so node E is the only one, which has not lost its stamp. If this aggregate receives unknown charges, C-F(0.0, 0.0, 0.0, 0.0), the charge of node E can be set to -1, because it has not had a positive charge before. Altogether the NEUNET-3-aggregate would return zero-charges A, B(0.0, 0.0) in the reaction-phase. The aggregate can’t return a concrete charge-value, but it would be more instructive, if it returned probabilities for certain charges. In this example it will be more like for A to get a positive charge and for B to get a negative charge. In detail the probability for A to get a positive charge is ⅔ and for B it’s ⅓. To reach this functionality the charge is not limited to the values (+1, 0, -1) any more, it can have any value between +1 and -1. A detailed description of this probability based approach is given in [3].

5 The Selective Approach

The selective model has an extended view on this problem. It does not use the stamp for remembering patterns/charges any more. For this model it is more accurate to call the stamps "weights", which can get non-integer values between 0 and +1, being "mathematically compatible" with the probability-based interpretation of the charge. So the weight may be termed "the probability-based interpretation of the stamp". If a new pattern is presented, the network returns the most similar one like the probability-based model, but it remembers the new input at once. When a pattern is recognized, its weight in the network increases, otherwise the weight decreases. For the association of new or incomplete patterns not only its similarity is a factor, but also its weight. So it is more like for a pattern with high weight to be associated with any input than for a pattern with low weight. And in case of learning faulty inputs the weight will decrease by time, because they won't be represented frequently (to increase weight, the same faulty pattern has to be repeated of course), so its influence on association will decrease.

5.1 Selective NEUNET - The Model

The nodes of the model have two characteristics. On the one hand they have their charge, which keeps the current information (read from receptors) and on the other hand they have a positive weight which represents the history of charges kept by a certain node. The weights, as in most other models in this field, provide the memory function. The association process consists of three main steps:

1. Read the current pattern from the receptors and propagate the charge through the whole network.
2. Adjust the nodes' weighs to remember the input.
3. If the pattern is unknown or incomplete, then try to find the most similar known one and load it.

For the interpretation of the node's charge, the model uses the probability for the unit to get a positive charge. So a charge of +1.0 corresponds to the probability of +1.0, 0.0 corresponds to the probability of +0.5, -1.0 corresponds to the probability of 0.0 and so on. The learning rate of the model decreases with every new pattern, because the weight/importance of a new pattern depends on the number of already known patterns of course. If an aggregate has leaned three patterns, for example "++", "+-", "--" (like the probability-based model before) and additionally a pattern, that is already known, "+-" for instance, then it ends up in the following weights: C-F (0.25, 0.5, 0.0, 0.25). In this case the probability for a positive charge in A is ¾ and ¼ for B (in contrast to ⅔ and ⅓ for A and B in the probability-based model). In Addition to the functionality of the probability-based model, the selective model pays attention to the repetition of known patterns.

5.2 Examples

Experiments with the new selective extension of NEUNET showed that the desired functionality of selective pattern association could be reached very well. Original

patterns are associated according to their weight in the network. The weight of information increases if it is presented to the network. In other case it decreases and if a pattern is not shown to Selective NEUNET long enough, it will be forgotten.

A selective NEUNET with 10x10 receptors has learned the digits form 1 to 5, a network with 1152 nodes and 238 aggregates was generated. If there is a faulty or incomplete pattern presented, the network tries to associate it to the most similar (and most important) original pattern.

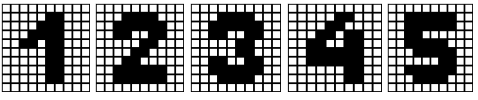


Fig. 6. Known patterns

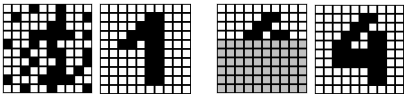
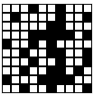
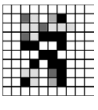
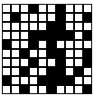
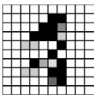
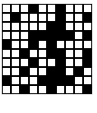
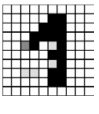
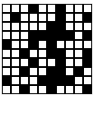
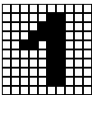


Fig. 7. Association of faulty and incomplete patterns. Left: input, right: output.

In the association process of NEUNET the most similar original pattern is reproduced gradually.

Table 2. Association of a faulty input

Step	Input	Output
1		
2		
...		
9		
10		

Previously the network learned new information by enhancing its network-structure, now the weights can be adjusted in addition. It enables the model to separate important information from a huge amount of any data.

6 NEUNET Compared to the Hopfield Model

As already mentioned NEUNET's functionality is very similar to the Hopfield network. Both models try to associate inputs with known patterns. Due to the fact that the network structure of NEUNET and Hopfield are totally different and the NEUNET- network can become very big (multiple layers of hidden nodes), Hopfield has an advantage in speed. Furthermore NEUNET does not always reach a stable state. For this reason it is essential to define a "maximum number of allowed steps". When this limit is reached, NEUNET stops the association process and shows the result. Experiments showed that a good limit is around "number of receptors / 2". However NEUNET has many advantages as well. It does not store complements of inputs, known patterns can be associated correctly within one oscillation and the orthogonality of patterns does not affect the association process. But the following table shows the main advantage of NEUNET – its precision.

Both networks (10x1 receptors) learned 10 patterns, followed by 500 patterns with increasing Hamming Distance were presented.

Table 3. Comparison NEUNET - Hopfield

Hamming Distance	NEUNET	Networks like Hopfield
0	100 %	100 %
1	91,4 %	33,0 %
2	66,6 %	13,3 %
3	28,6 %	2,4 %
4	15,0 %	0,2 %
5	2,0 %	0,0 %

Table 4. Advantages/disadvantages of NEUNET

NEUNET	Hopfield
Multiple layers of units	One layer
Dynamic network structure	Static network structure
Leaning by weight adjustment and structure enhancement	Leaning just by weight adjustment
No convergence in some cases	Convergence is proved
No complements stored	Complement of patterns automatically stored
Known patterns can be associated correctly within one oscillation	Correct association of known patterns not assured
Character of input patterns does not affect association	Patterns should be (nearly) orthogonal

One can see that the recognition rate of NEUNET is considerably higher than the rate of the Hopfield model. The following table recapitulates the advantages/disadvantages of the model.

7 Conclusion and Further Work

Nowadays neural networks are mainly used for solving classification problems (e.g. recognition of hand written texts, voice and speaker recognition and in many other cases). The intention of "selective" NEUNET which was introduced in this paper was to have a network, which is able to separate important from unimportant information autonomously and to learn that without a supervisor. Our theoretical considerations and experiments showed that this expected behaviour was reached. Our future work will concentrate on testing this network in real world applications (e.g. simple games) and to try to enhance it for serial associations like NEUNET3S [6].

References

1. Reichl, E.R.: Neuronennetz-ähnliche Strukturen als Assoziativspeicher. In: Applied Computer Science, vol. 8, Datenstrukturen, Graphen, Algorithmen (1978)
2. Borgelt, C., Klawonn, F., Kruse, R., Nauck, D.: Neuro-Fuzzy-Systeme. Vieweg Verlag (2003)
3. Andlinger, P.: FUZZY-NEUNET: A non standard neural network, Artificial neural networks. In: Prieto, A.G. (ed.) IWANN 1991. LNCS, vol. 540, Springer, Heidelberg (1991)
4. Hopfield, J.: Neural networks and physical Systems with emergent collective computational abilities. In: Proc. of the National Academy of Sciences, USA, vol. 79 (1982)
5. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representations by Error Propagation. In: Rumelhart, D.E., McClelland, J.L. (eds.) Parallel Distributed Processing, vol. 1, MIT Press, Cambridge, MA (1986)
6. Andlinger, P.: Serielle Assoziation in neuronennetzähnlichen Netzwerken. In: Proc. Of the 33rd International Scientific Colloquium, Ilmenau, Germany (1988)

Towards Constraint-Based Preservation in Systems Specification

Thomas Triebsees and Uwe M. Borghoff

Universität der Bundeswehr München
85577 Neubiberg, Germany
{Thomas.Triebsees,Uwe.Borghoff}@unibw.de

Abstract. Model transformation has applications in many areas and usually is to *preserve* certain properties. In model-driven software development, e.g., abstract models are transformed into more specific ones while preserving the behavior of the overall specified system. In this paper we demonstrate how constraint-based preservation, which we have initially developed in the context of digital archiving, can be applied to systems specification. We demonstrate how our approach helps to preserve communication flows inside a component-based system. In particular, formal *preservation constraints* allow us to prove invariance of certain significant properties of a component interaction specification if the transformation process adheres to a given set of such constraints.

1 Introduction and Motivation

Nowadays, computer-based systems are the key element in many application domains. As a central challenge, complexity and domain-specific characteristics of such systems have to be handled by modern software and hardware engineering methods. Our initial research in the field of digital archiving has shown that digital archives constitute a challenging application domain in this respect. Usually, digital archives manage large bodies of digital data containing diverse kinds of information and having different types of (semantic) relationships. For this purpose, archives mostly comprise a complex hardware and software environment. The crucial tasks of the underlying components is to ensure the preservation of digital material over a long (i.e., an essentially unspecified period of) time. The software and hardware architecture is therefore often highly modularized so as to facilitate component exchange when technology evolves. The interested reader is referred to the reference model for Open Archival Information Systems (OAIS) [1] to get an impression of the complexity of archiving systems.

Technology evolution or changes in preservation requirements regularly necessitate component migration. What was true for document migration also holds for component migration, namely that they usually *preserve* certain invariants [2], [3], [4], [5], [6]. As a major difference compared to documents, components may communicate with each other and adhere to an internal workflow. A typical preservation task in this scenario is that the external view on a component remains unchanged while performing internal changes.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 894–902, 2007.
© Springer-Verlag Berlin Heidelberg 2007

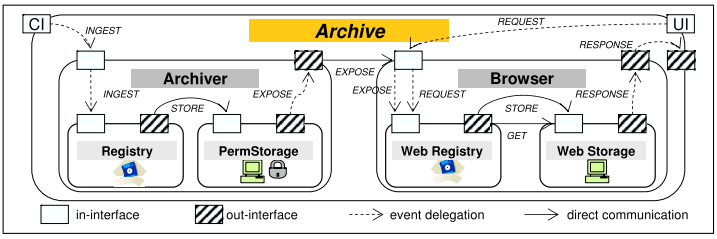


Fig. 1. Example web archiving system

In [7] we have extended our preservation framework by a so-called automata concept. This concept is a general approach to semantically evaluate *and* construct graph queries that conform to a regular grammar. The graph structures are specified using first-order logic. We use techniques from model checking by constructing suitable query automata. So far, the extension of our framework has mainly been motivated by document engineering examples like link-consistency in web-documents. In this paper we show how this extended preservation framework can be used in the context of system specifications. As case study we recall the example of [8], which stems from digital archiving. We informally introduce excerpts of the specification of the communication flow inside a web-archiving system. The system consists of several components that are hierarchically structured and can communicate using a set of pre-defined events. Our task is to introduce a logging component that logs occurrence of a subset of these events. This change, however, is constrained in the sense that the general communication flow must not interrupted by the logging component. Properties of this kind can be challenging to guarantee in large specifications, such that automated constraint checking is surely desirable. We use our preservation language to formally specify the transformation task as a whole as well as the constituent requirements.

2 The Scenario - Communication Inside a Web Archive

Fig. 1 shows our example web archive in the fashion of UML component diagrams. It consists of seven components. The *Archiver* has a permanent storage for long-term preservation of the hosted websites. The servers being annotated with locks indicate that access to this storage environment is not permitted from "outside" the archive. The latter is provided by the *Browser*-component, which includes a web-storage with fast server access. The archive has two external communication interfaces – the customer interface (*CI*) and the user interface (*UI*). Customers can ingest a website to the archive, which corresponds to the event *INGEST*. This event then is delegated to the *Archiver* and *Registry* components, where the latter sends a *STORE* event to the permanent storage. The *Browser* component provides a service for full website browsing as well as quick search facilities. Users can request these services by issuing a *REQUEST*. Upon request,

they will receive their data as a *RESPONSE*-event. Our task is to extend the archive by an additional component *Log* that serves as an intermediate instance and logs the events *INGEST*, *REQUEST*, and *RESPONSE*. We require the introduction of the *Log* component to be liberal in the sense that it preserves the external view on the archive's communication flow as described above.

3 Basic Model

In the following we want to show how our preservation framework of [9,7] can be used to model the properties in question and specify the preservation task. In order to get a feeling of how we employ our framework to guarantee adherence to specified constraints, we first briefly recall the state-based transformation semantics of basic operations. After that, we model component interaction specifications as a prerequisite to model archival communication flows in Sect. 4.1.

State-based transformation model. Fig. 2 shows the basic transformation model. An archive syntactically comprises a set of object types $\mathcal{T}$ with subtyping $\leq$, an infinite set of object IDs O, and function symbols F that can be executed on the objects. The signature is interpreted in an algebra, the latter constituting a system state. As in reality, states can be changed by activities like, e.g., creation of an object. Hence, we see the entity model as an algebra that evolves over time, which corresponds to the concept of *Abstract State Machines* (ASM, [10]). We support three *basic operations* that can cause state changes: creation, transformation, and deletion of an object. Model transformations are modeled as sequences of basic operations. In Fig. 2, the algebra $\mathcal{A}'$ is generated from $\mathcal{A}$ by creating $o5$, the latter having $o3$ as an attribute.

Component and specification model. In our scenario, we specify all necessary entities by four different object types as is shown in Fig. 2. Specifications are modeled by type *Specification* having three attributes – *components* (the set of components belonging to this specification), *events* (the set of specified events), and *eventDels* (a set of event delegation specifications as indicated in Fig. 1). Events are modeled by type *Event*, which comprises an *id*-attribute. *Components* have a *name*, an interface *inInterface* for incoming events (listed by the corresponding event's *id*), and an interface *outInterface* including all

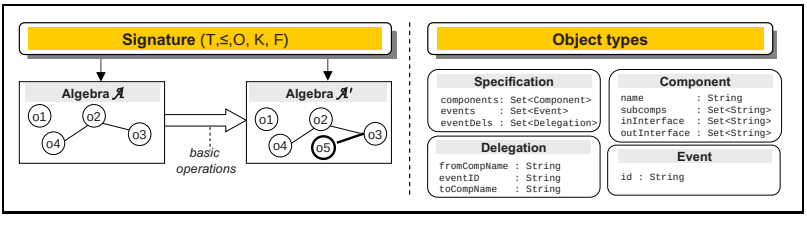


Fig. 2. Archive model

events that can be issued by the component. Component structure is reflected by the *subcomps*-attribute, which lists all component names that are *directly* below the corresponding component in the hierarchy.

4 Property Preservation

In this section we formalize extraction of some key properties from component interaction specifications, namely component hierarchy and event interactions. After that we provide a set of preservation constraints that, if adhered to, guarantees an unchanged external view on the communication after transformation.

4.1 Specifying Significant Properties

As already mentioned in Sect. 2, our preservation focus lies on two properties when extending the specification. First, the component hierarchy must not change in a manner that sub-component relationships that were true before transformation do not hold after transformation. Second, we want to preserve the following: If two components can interact using an event e, this interaction is still possible after the specification extension. Analogous to [9], we use so-called concepts to express these formal properties. Fig. 3 depicts the concept $\mathcal{K}^{subc}$, which specifies the subcomponent-relationship.

The concept interface shows that subcomponent-relationships are evaluated for a specification x_0 and two components x_1, x_2. The concept has exactly one context which specifies the semantic property that must hold in order to satisfy the overall concept. Formally concept satisfaction is expressed by

$$\mathcal{A}, \eta \models \mathcal{K}(o_0, ..., o_n)[C] \Leftrightarrow C \in C_{\mathcal{K}} \wedge \mathcal{A}, \eta[o_0/x_0, ..., o_n/x_n] \models \iota_C.$$

Given a variable substitution η and objects $o_0, ..., o_n$, the concept $\mathcal{K}$ holds in context C for these objects, if C is a context of $\mathcal{K}$ and the embedding formula ι_C is valid for the appropriately updated variable substitution.

The second property – component interaction using an event e – would be laborious when solely expressed in first order logic. Fortunately, our automata concept is a remedy. It allows us to query a graph structure that is specified inside a first-order theory. To be more precise, we 1) specify a graph structure in which objects constitute vertices, 2) specify a regular query language, and 3) construct a query automaton that evaluates words of the query language in

Concept interface KI^{subc}**:** $\langle x_0 : \alpha_0, x_1 : \alpha_1, x_2 : \alpha_2 \rangle$	
	$\alpha_0 \leq Specification \wedge \alpha_1 \leq Component \wedge \alpha_2 \leq Component$
Context C^{subc}**:**	
$\iota_{C^{subc}}$	$x_1 \in components(x_0) \wedge x_2 \in components(x_0) \wedge name(x_1) \in subcomps(x_2)$
Concept $\mathcal{K}^{subc}$**:**	
$KI_{\mathcal{K}^{subc}}$	KI^{subc}
$C_{\mathcal{K}^{subc}}$	$\{C^{subc}\}$

Fig. 3. Specification of the concept $\mathcal{K}^{subc}$

the specified graph structure. With objects being vertices in the queried graph structure, we have a natural semantics for query results. We refer the reader to [7] for details of the formal constructions.

In our case, the query language comprises all non-empty sequences of letters and digits. Formally, it adheres to the following grammar G_{query}:

$$
\begin{array}{ll}
EventID & = Letter\ EventID \mid Digit\ EventID \\
Letter & = \texttt{a} \mid \ldots \mid \texttt{z} \mid \texttt{A} \mid \ldots \mid \texttt{Z} \\
Digit & = \texttt{0} \mid \ldots \mid \texttt{9}
\end{array}
$$

The result of the graph structure specification is depicted in Fig. 4. It directly reflects the component interactions shown in Fig. 1. Indicated on the upper part of Fig. 4 is the formal specification of the structure. This specification comprises a specification of the vertex set and a specification of the edge set [7] and is partly listed in the following table.

Vertex constraint ϕ_v	
$x \in components(s)$	
Label constraint ϕ_l	**Edge constraint** ϕ_e
$l \in intersect(\ outInterface(x_0),\ inInterface(x_1)\ )$	$\exists c : Component \bullet (x_0 \in subcomps(c) \wedge x_1 \in subcomps(c))$
$l = name(e)$	$name(x_0) \in subcomps(x_1) \wedge$ $\exists d \in eventDels(s) \bullet (\ fromCompName(d) = name(x_0) \wedge$ $toCompName(d) = name(x_1) \wedge$ $eventID(d) = name(e)\ )$

The vertex specification is read as follows: An object o is a vertex in the component interaction graph of a specification *spec*, iff ϕ_v holds for the variable assignment $\eta := \{o/x, spec/s\}$. According to ϕ_v in the table, all components that are listed in the *components*-attribute of a specification belong to the component interaction graph of that specification.

Edge specifications are read as follows: There is an edge with label *lab* between objects o_0 and o_1 in the component interaction graph of *spec*, iff for at least one combination ϕ_l and ϕ_e of label and edge constraints $\phi_l \wedge \phi_e$ holds for the variable assignment $\eta := \{o_0/x_0, o_1/x_1, lab/l, spec/s\}$. The first row of the edge specification in the table above generates an edge with label $eventID$ between two components, iff both components have the same super-component and the $eventID$ is listed in both the out-interface of the source component and the in-interface of the target component. This corresponds to direct communication as, e.g., takes place between the *Archiver* and *Browser* component using the $EXPOSE$-event. The second edge specification from above expresses upwards event delegation between the out-interfaces of a sub- and a super-component. In our example, the *Browser*-component delegates the $RESPONSE$-event from the out-interface of the $WebStorage$-component to the *Browser*'s out-interface. Downwards event delegation is specified similarly.

Since the query language conforms to a *regular* grammar, we can construct a kind of product automaton $A^{\times}(G_{query}, Graph_{spec})$ [7] that accepts exactly those sequences of *eventIDs* that are valid paths in the queried graph. In Fig. 4 the query `INGESTINGEST`, e.g., computes the state o_{17} (which is the object ID of the

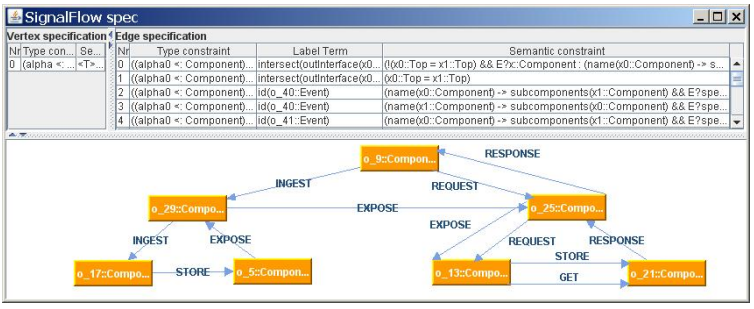


Fig. 4. Component interaction graph

Registry in our example) if started at state o_9 (*Archive*). Using this automaton and an appropriate acceptance condition, we can express the concept

$$\mathcal{K}^{comm}(spec, c_s, e, c_t) \Leftrightarrow$$
$$A^{\times}(G_{query}, Graph_{spec}) \text{ has a run from } c_s \text{ to } c_t \text{ computing a word in } id(e)^{+}$$

the full definition of which we skip due to space limitations. The acceptance condition of $\mathcal{K}^{comm}$ exactly expresses the property that the specification *spec* facilitates the source component c_s to communicate the event e to the target component c_t. The communication path may include direct communication or event delegations as specified above. Notice that all these concept and graph specifications scale well even for large component interaction structures.

4.2 Specifying Property Preservation

When introducing the *Log* component, we have to extend (i.e., to transform) the original specification, which may produce undesired results when executed in an uncontrolled manner. After having specified some significant properties of component interaction, let us now switch to the preservation thereof. For this purpose, we use the preservation language introduced in [9] to constrain the transformation process appropriately. We firstly list the formal constraint set Ξ. In a second step we explain their exact meaning.

(1) $\forall c : Component \bullet \mathtt{pres}_o(c \mapsto Component, c[Component\{name\}])$
(2) $\forall e : Event \bullet \mathtt{pres}_o(e \mapsto Event, e[Event])$
(3) $\forall s : Specification, c_s : Component, c_t : Component \bullet$
$\mathtt{pres}_k(\{s \mapsto Specification\}, \mathcal{K}^{subc}(s, c_s, c_t))$
(4) $\forall s : Specification, c_s : Component, c_t : Component, e : Event \bullet$
$\mathtt{pres}_k(\{s \mapsto Specification\}, \mathcal{K}^{comm}(s, c_s, e, c_t))$
(5) $aSpec \mapsto Specification$

The first two constraints represent object preservation constraints, where the first one, e.g., assures that, whenever a component c is transformed to another component $\delta(c)$ ($c \mapsto Component$), the name of $\delta(c)$ equals the name of c ($c[Component\{name\}]$). The type $Component\{name\}$ is an abstraction

of type *Component* having only the *name*-attribute. Constraint satisfaction $(\mathcal{A}, \Delta, \mathcal{A}') \models \mathtt{pres}_o(o \mapsto \tau, o[\tau])$ is defined w.r.t. a preceding and a subsequent state and a transformation sequence Δ containing basic operations. The full semantics is given in [9]. Notice that we do not care about the $inInterface$ or $outInterface$-attributes so far. From the graph specification of the last section one can already guess that preservation of communication flows adjusts the communication interfaces of the transformed components accordingly.

Constraints (3) and (4) represent concept preservation constraints. Constraint (3) assures that, whenever c_s is a sub-component of c_t in s $(\mathcal{K}^{subc}(s, c_s, c_t))$ and s is transformed to another specification object, c_s and c_t are transformed as well and $\mathcal{K}^{subc}$ holds for the transformed objects $\delta_1(s), \delta_2(c_s), \delta_3(c_t)$. Constraint (4) assures preservation of communication flows as explained in the last section.

Finally, the fifth constraint assures that our example specification object *aSpec* is transformed to type $Specification$. This type of constraint is satisfied iff the transformation sequence contains a corresponding transformation.

The advantages of this rigorous approach become apparent when we want to gain confidence in the specification and prove certain properties of the result system. In the following, we prove an example and, thus, show the typical proof style applicable to the type of constraints provided here.

Lemma 1
If $\mathcal{A}, \eta \models name(c_s) = n \wedge \exists c_t : Component, e : Event \bullet \mathcal{K}^{comm}(aSpec, c_s, e, c_t)$ *and* $(\mathcal{A}, \Delta, \mathcal{A}') \models \Xi$*, then* $\mathcal{A}', \eta \models \exists c'_s \in components(\delta(aSpec)) \bullet name(c'_s) = n$.

Proof. If $(\mathcal{A}, \Delta, \mathcal{A}') \models \Xi$, then in particular constraints (5), (4), and (1) are satisfied. Constraint (5) assures that *aSpec* is transformed by some transformation δ. From (4) we know that communication flows are preserved. In particular, $\mathcal{K}^{comm}(aSpec, c_s, e, c_t)$ holding together with (3) imply that there are transformations $\delta_1(c_s), \delta_3(e), \delta_4(c_t)$ such that $\mathcal{A}' \models \mathcal{K}^{comm}(\delta(aSpec), \delta_1(c_s), \delta_3(e), \delta_4(c_t))$. Constraint (1) assures $name(\delta(c_s)) = name(c_s)$, i.e., we have $\mathcal{A}' \models \exists c'_s : Component \bullet name(c'_s) = n$ and we remain to show that this component is in the component set of $\delta(aSpec)$. This can be seen from the graph specification underlying $\mathcal{K}^{comm}$. In each of the edge constraints, a subcomponent-relation is pre-required. Hence, there is a direct super-component or sub-component of c_s. Constraint (3) from above assures preservation of this relation. We conclude the result for the case where c_s has a super-component. The other result follows analogously. If c_s has a super-component c, then $\mathcal{K}^{subc}(aSpec, c_s, c)$ holding implies $\mathcal{K}^{subc}(\delta(aSpec), \delta_1(c_s), \delta_5(c))$ for a new transformation δ_5. The implementing formula of $\mathcal{K}^{subc}$ finally assures $\delta_1(c_s) \in components(\delta(aSpec))$ (cf. Fig. 3).

Informally spoken, this lemma states that we do not need a separate preservation constraint for component containment in a specification. If models and constraint sets grow larger, efficiency becomes a crucial property. Avoiding unnecessary additional constraints clearly contributes to increasing efficiency.

5 Constraint Checking Results and Future Work

We have prototypically implemented a system that facilitates to specify significant object properties and preservation requirements of the fashion described in this paper. For the running example, we have provided the component interaction specification in an XML-format. This specification is parsed and appropriately transformed into formal objects. Transformations run inside an own environment such that the system is aware of newly created or transformed objects. We have tested our example with different incorrect and correct specification extensions. Indeed, constraint satisfaction has been recognized correctly, evaluation time was no issue in this small example. Future work surely will cover system tests with more sophisticated examples. In particular, type checking of formulae and construction of product automata takes time. However, models are created iteratively. All specified graphs and automata are re-constructed only *partly* on state change, if necessary.

Another part of future work will cover automated model construction. In our example, preserving the component hierarchy gives rise to a basic target model from which to start model completion while respecting specified constraints. If major properties are graph-based, the model completion process can run along graph paths and adjust object attributes as required by the graph specifications and preservation constraints. That way the $inInterface$ and $outInterface$-attributes in our example, e.g., can be adjusted automatically.

References

1. Consultative Committee for Space Data Systems: Reference model for an open archival information system. Technical report, Space Data Systems (2002)
2. Margiara, T., Steffen, B.: Backtracking-free design planning by automatic synthesis in metaframe. In: Astesiano, E. (ed.) ETAPS 1998 and FASE 1998. LNCS, vol. 1382, pp. 188–204. Springer, Heidelberg (1998)
3. Brim, L., Černá, I., Vařeková, P., Zimmerova, B.: Component-interaction automata as a verification-oriented component-based system specification. SIGSOFT Softw. Eng. Notes 31(2), 4–11 (2006)
4. McBrien, P., Poulovassilis, A.: A uniform approach to inter-model transformations. In: Proc. 11th Int. Conf. on Advanced Inf. Sys. Eng (CAiSE 1999), pp. 333–348 (1999)
5. Szemethy, T., Karsai, G., Balasubramanian, D.: Model transformations in the model-based development of real-time systems. In: Proc. 13th IEEE Int. Symp. on Eng. of Comp. Based Sys (ECBS 2006), pp. 188–196. IEEE Computer Society, USA (2006)
6. Madhavan, J., Bernstein, P.A., Domingos, P., Halevy, A.Y.: Representing and reasoning about mappings between domain models. In: 18th Nat. Conf. on Artificial Intelligence, American Association for Artificial Intelligence, pp. 80–86 (2002)
7. Triebsees, T., Borghoff, U.M.: Towards automatic document migration: Semantic preservation of embedded queries. In: Proc. of the Int. ACM Symp. on Doc. Eng (DocEng 2007), ACM Press, New York (2007)

8. Triebsees, T., Borghoff, U.M.: Towards a theory for preservation in the field of system specification with focus on digital archives. In: Quesada-Arencibia, A., et al. (eds.) Proc. 11th Int. Conf. on Computer-Aided System Theory (Eurocast 2007), IUCTC, pp. 303–304 (2007)
9. Triebsees, T., Borghoff, U.M.: A theory for model-based transformation applied to computer-supported preservation in digital archives. In: Proc. 14th Ann. IEEE Int. Conf. on the Eng. of Comp. Based Systems (ECBS 2007), IEEE Computer Society Press, Los Alamitos (2007)
10. Börger, E., Stärk, R.: Abstract State Machines. A Method for High-Level System Design and Analysis. Springer, Heidelberg (2003)
11. Cheney, J., Lagoze, C., Botticelli, P.: Towards a theory of information preservation. Technical report, Cornell University, Ithaca, New York (2001)

Automated Re-establishment of the Synchronization of Real and Simulated Project Workflow

Manfred Mauerkirchner and Gerhard Hoefer

Austrian Partner of International Universities, A-4060 Leonding, Austria
m.mauerkirchner@htl-leonding.ac.at
g.hoefer@htl-leonding.ac.at

Abstract. It is among the main responsibilities of each managing instance to maintain control on project workflows. To be successful in this aspect we propose a set of systems that guarantee the emulation of any project workflow at any time. We concentrate on humans as project resources characterized by individual profiles and a specific methodology to express heuristic relationships among all involved persons. We use common and optimized project plans to ensure that real project workflow and its real time simulation run synchronously. The most challenging task is to keep both running systems in track although many different external disturbances may occur. Our specific work tries to automate or at least to support automation of all these processes by providing a specific framework.

1 Introduction

The overall intention of the presented work is to automate the project control process of the concerted activities of different project groups within one project. At least two processes have to be observed and synchronized simultaneously: The real project flow and its real-time emulation, both using a common, permanently adapted and optimized project flow plan. Therefore, such optimized project flow plans, consisting of an arbitrary number of project activities of various types and based on a pool of human resources characterized by specific profile parameters, should be available in order to predict the real project flow. These profile parameters describe the individual characteristics of a human resource like the various skills and costs, and collective characteristics like the heuristic relationships among all members of the resource pool. The individual characteristics are scalar valued, whereas the collective characteristics consist out of a matrix scheme containing all mutual relationships as matrix elements.

The related project activities with estimated duration are grouped in certain project subtasks and are sequentially processed by individual project groups whose members stem from an arbitrary set of non specific human resources. The mentioned subtasks are nodes in a directed graph defining a predecessor-successor relation between subtasks. (see Fig.1)

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 903–911, 2007.
© Springer-Verlag Berlin Heidelberg 2007

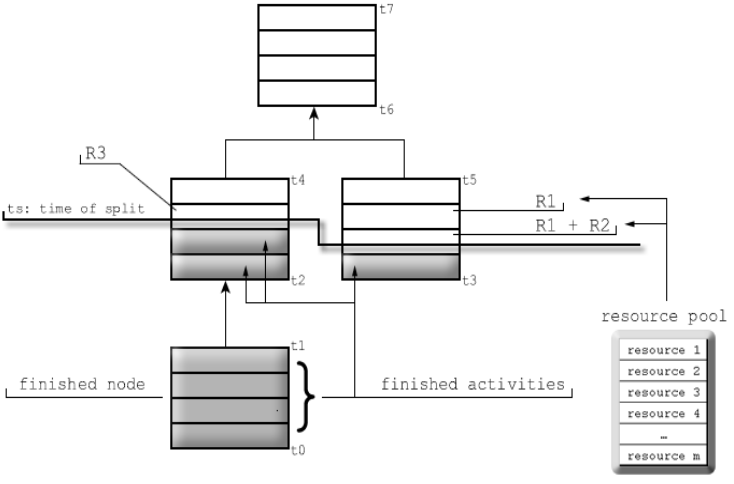


Fig. 1.

The termination time of each activity is marked by an individual milestone as common in project working. These marks existing in the simulated and in the real project flow plan are crucial for the further assessment of the synchronicity of both project flow plans.

2 Methodologies and Systems

In the four subsections below the necessary core methodologies for establishing a project flow plan and their systemic integration will be introduced. This introduction will be done in a project relevant chronological order:

In the first subsection the base planning subsystem and its three core methodologies will be discussed. The next subsection deals with the kinds of disturbances that may arise during the execution of project flow plans based on incorrectly estimated profile parameters. Two core methodologies and their related subsystems to overcome such difficulties will be discussed in the following subsection. In the concluding subsection the necessary steps to re-establish synchronicity in real and simulated project flow are described step by step in a more detailed way.

2.1 Base Planning Subsystem

Our planning subsystem consists of three parts each using another heuristic methodology: The *Local Optimizer* searches optimal project groups within a dynamic environment for activity-specific requests and using appropriate genetic algorithms.[4] The *Flow Simulation* simulates one project work flow based on the directed project graph using the Local Optimizer. The *Global Optimizer*

searches for the optimal *project flow plan* by employing a numeric non-gradient method introduced by Rosenbrock.[7]

Motivation for these heuristic methodologies: The very high degree of complexity of workflow planning problems renders the usage of exact solution methods practically irrelevant. What is preferred nowadays is the implementation of heuristic methods to generate solutions iteratively by checking so-called priority rules and, in addition, using adaptive random sampling methods.[1]

As a result all project plans contain a complete workflow of project activities and schedules of all involved persons. The most important requirements that such a planning system must fulfil are (see also [5][6]):

- calculation of complete project plans very efficiently and whenever needed
- re-planning of partly executed plans (from any moment on)
- re-planning of partly executed plans with regard of additional restrictions: i.e. existing group memberships remain unchanged
- delivery of schedules/plans between two arbitrary dates

As a preliminary general restriction we have to commit that all estimated duration stem from an external expert system and are regarded as constant without restriction of generality.

2.2 Category-1 and Category-2 Problems

Based on a common actual project plan, both project workflow systems (*Real Flow System* and its associated *Simulated Flow System*) are active and running synchronously. The workflow in the real project becomes emulated and visualized by the so-called *Controlling Subsystem*, used by the project manager in charge (see Fig.2).

Various problems may occur during the processing of the planned project activities that could be grouped in two main categories:

Category-1: External and minor sized internal disturbances. Category-1 problems arise when external disturbances like allocation or deletion of human resources or minor sized deviation from pre-calculated time intervals of already finished project activities occur at any moment (*time of split*, see Fig.1). Minor sized deviations should be caused by some isolated miss-judged skill and interaction parameter values. Nevertheless, the project manager is able to cope with the latter situation since his assessment of the individual members the affected project group and their respective profile parameters enables him to re-judge the crucial parameter values especially in the case of only very obvious patterns of deviations. Thus, disturbances are regarded as minor sized if no complex pattern of miss-judged profile parameter values is exhibited and therefore only isolated parameter values have to be changed.

Category-2 : Medium and major internal disturbances. Category-2 problems arise from incorrectly judged profile parameters concerning the personal skills of project members and the mutual relationships within project groups.

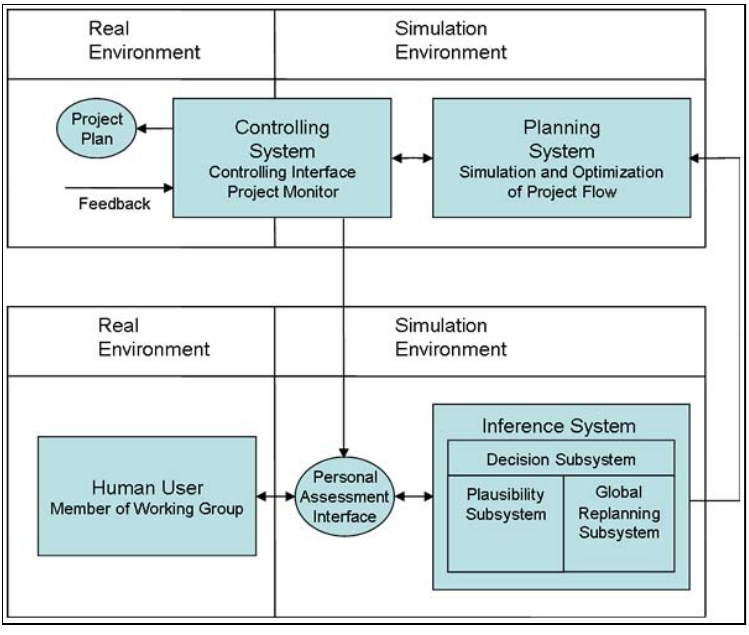


Fig. 2.

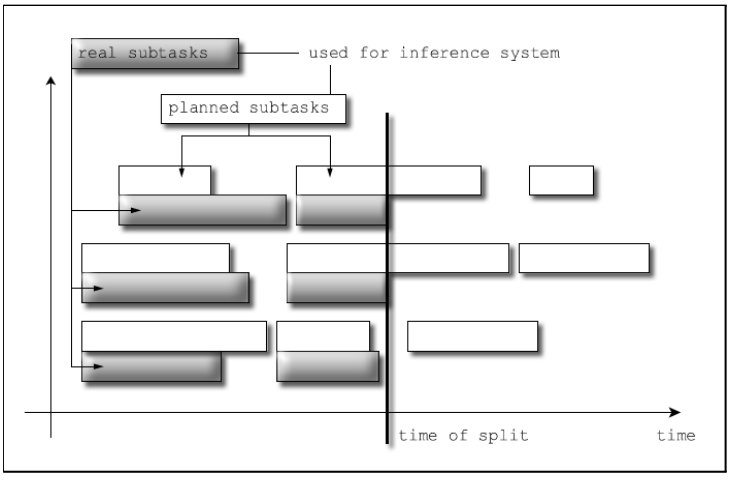


Fig. 3.

These skill and group interaction parameters are crucial for the synchronization of the real and the simulated project flow. These disturbances become evident in increasing inconsistencies concerning the real project flow milestones and the associated simulated milestones (see Fig.3). In that case the pattern of deviations

is too complex to be re-judged by the project manager just on the basis of his assessment of individual members of the affected project group and their respective profile parameters.

Disturbances may be regarded as medium sized if they are caused by a confined set of miss-judged skill parameters values and some isolated interaction parameter values that also have to be adjusted. Major sized disturbances are caused by arbitrary sets of miss-judged skill and interaction parameters. Therefore, the degree of disturbance should always be judged on the basis of its underlying profile parameter values. Only realistic assumptions of the values of their individual profile parameters keep both flows in track permanently. Consequently, heuristic methods are employed to synchronize the flow of the real and the simulated activity related milestones. Synchronized project flows must show only minor deviations in execution times of the real and their associated simulated milestones. Two separate strategies have already been proposed to overcome category-1 and category-2 problems.

2.3 Solutions

Category-1 problems are dealt with by an already running prototype published first in [3]. In the case of minor deviations from pre-calculated time intervals of already finished project activities the project manager may be allowed to readjust obviously deviating profile parameters on purpose. If these adjusted values of profile paramteres do not suffice to re-establish synchronocity in further runs of the project work flow the project manager has to employ the inference system.

Category-2 problems are treated by a strategy introduced in [5]. The idea behind this strategy is the insertion of two components into the overall system that are able to induce the ability of self learning. These new components are the personal assessment interface which is able the collect suggestions for improvement of both skill and group interaction parameters from all members of a concerned group and the inference system which tries to gradually improve the tuning of these system inherent parameters by iteratively incorporating the collected suggestions for improval into the actual set of parameter values (see Fig.2).

Realization as Bipartite Inference System. A bipartite version of the inference system will be able to deal with various degrees of medium to major disturbances of the project flow in two different ways: Medium sized disturbances are corrected by a non-heuristic search methodology based on a multivalent tree structure[2], whereas major disturbances are corrected by a heuristic methodology based on genetic algorithms as described in [5].

Multivalent tree structure: This data structure is created as one overall tree structure containing all these suggested values for all miss-judged skill parameters as well as the values of obviously correctly judged skill parameters. The tree is made up of all suggestions for adjustments of skill parameters ordered in an hierarchical way as common in this data structure. Vertices of one tree layer contain all suggestions for one particular profile parameter. This tree is based on a list structure made up of all suggested profile parameter values (see Fig.4).

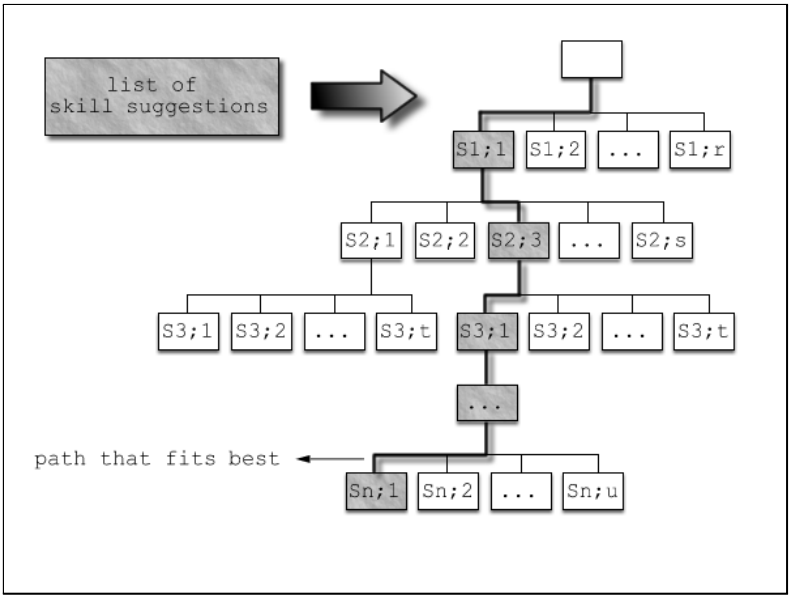


Fig. 4.

Branches of the Bipartite Inference System: The tree-based inference subsystem is realized as the plausibility subsystem whereas the GA-based inference subsystem is realized as the global re-planning subsystem. These subsystems are regarded as the two main branches of the bipartite inference system. One further subsystem of the inference system is the socalled decision interface which initiates the activation of one of the branches introduced above.

Decision interface. The project manager has to determine via the decision interface which branch of the bipartite inference system needs to be activated. This decision is based on the fact whether the disturbance is regarded as medium or major sized according to the actual suggestions for improval. As mentioned above medium sized disturbances are corrected by the plausibility subsystem whereras major sized disturbances are corrected by the global re-planning subsystem.(see Fig.2).

In order to introduce the new branch of the bipartite inference system, we are going to describe the complete process of profile parameter adjustment for the case of medium sized project flow disturbances (see Fig.5).

2.4 Synchronization of Environments

The synchronization process can be described executing the following steps:

Step 1 : Suggestions for adjustment of potentially incorrect profile parameters have been collected by the personal assessment interface of all members of

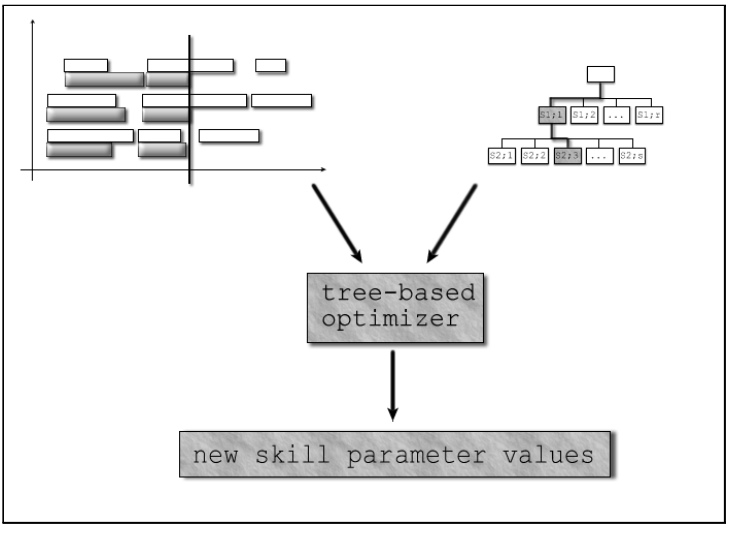


Fig. 5.

affected project work groups in order to re-establish synchronicity of real and simulated project flow. These individual suggestions are visualized on a control monitor and checked for reasonableness by the project manager. All reasonable parameter values a stored at a central database for further processing .

Step 2 : The project manager has to decide which way to adjust the values of the affected profile parameters has to be employed. This decision is based on the degree of actual disturbance which is evident in the number of profile parameters that are suggested for adjustment. Since we want to introduce the tree based search system from now on only this branch of the inference system will be discussed.

The project manager uses the decision interface to initialize and start the plausibility subsystem. During this phase of initialization all reasonable suggestions for adjustments of profile parameters are transferred to that subsystem.

Step 3 : In case of medium sized disturbances the miss-judged values of isolated interaction parameters have to be corrected by the project manager based on the manager's experience gained from former similar project work. Therefore, the project manager has to choose the most convinient values for all affected interaction parameters.

Step 4 : The plausibility subsystem has to find a path in the profile parameter tree whose individual parameter values are able to re-synchronize the real and the simulated project flows. The real project flow is characterized by a sequence of milestones of already finished project activities. Therefore, re-simulation of

these past activities, carried out with all paths contained in the tree, should detect one path that yields a considerably improved result.

In order to determine the most plausible profile parameters we have to find the path that yields the minimum deviation of activity milestones in the real project work flow and the associated milestones in the simulated project work flow :

- Re-plan the project flow utilizing the profile parameters of one path of the multivalent tree.
- Determine the differences in time of each already executed milestone in the real project flow and its associated milestone in the simulated project flow.
- Determine the sum of the absolute values of all these differences.
- Repeat this procedure for all paths of the tree and find the minimum.
- Check whether this minimum predicts the real project flow in an acceptable way.
- If not start the global re-planning subsystem.

3 Summary and Preview

In this work five core methodologies have been introduced in order to plan and execute project work flow in an automated way. These core methodologies are:

- the Local Optimizer
- the Flow Control
- the Global Optimizer
- the Tree Based Optimizer
- the Global GA Based Optimizer

Core methodologies are grouped in three different systems that are employed in characteristic situations during planning and executing a project:

- the Base Planning System, employing Local Optimizer, Flow Control and Global Optimizer
- the first part of the Inference System (Plausibility Subsystem) dealing with medium sized disturbances within the project flow, employing Local Optimizer, Flow Control and Tree Based Optimizer
- the second branch of the Inference System (Global Replanning Subsystem) dealing with major sized disturbances within the project flow, employing Local Optimizer, Flow Control, Global Optimizer and Global GA Based Optimizer

A first prototype proving a possibility of execution of the Plausibility Subsystem has already been implemented. This work was done mainly by two students (Martin Enzelsberger, Gerald Kettlgruber) concerning their specific diploma thesis at the College of Data Processing and Organization at HTBLA Leonding (2007).

References

1. Dawson, C.W.: A Dynamic Sampling Technique for the Simulation of Probabilistic and Generalized Activity Networks. Omega-Int. Journal Mgmt. Sci. 23(5) (1995)
2. Luger, G.F.: Artificial Intelligence: Structures and Strategies for Complex Problem Solving. Addison-Wesley Longman, Amsterdam (2002)
3. Mauerkirchner, M.: Event Based Modelling and Control of Software Processes. In: Engineering of Computer-Based Systems - Proceedings ECBS 1997, Monterey, California (1997)
4. Mauerkirchner, M.: A General Planning Method for Allocation of Human Resource Groups. In: Moreno-Díaz Jr., R., Buchberger, B., Freire, J.-L. (eds.) EUROCAST 2001. LNCS, vol. 2178, Springer, Heidelberg (2001)
5. Mauerkirchner, M., Hoefer, G.: Towards Automated Controlling of Human Project Working Based on Multiagent Systems. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, Springer, Heidelberg (2005)
6. Michalewicz, Z., Fogel, D.B.: How to Solve It: Modern Heuristics. Springer, Heidelberg (2002)
7. Rosenbrock, H.H.: An Automatic Method for Finding the Greatest or Least Value of a Function. Computer Journal 4 (1960)

Automated Modelling in Empirical Social Sciences Using a Genetic Algorithm

Bernd Brandl

University of Vienna, Department of Industrial Sociology, Faculty of Business, Economics and Statistics, Bruenner Strasse 72, A-1210 Vienna, Austria
bernd.brandl@univie.ac.at

Abstract. Automated modelling is of increasing relevance in empirical social sciences because of the increasing availability of potentially important variables. The availability of many variables causes uncertainty about which variables should be included in parsimonious models for the explanation of phenomena in social sciences and which variables should be excluded. Given a large number of potentially informative variables this paper argues that the use of genetic algorithms for Bayesian model selection allows the efficient automated identification of an optimal subset of variables. The advantages of using a genetic algorithm as a method for automated modelling is exemplified by the identification of previously unknown but important causal relationships for long-run inflation, the share spent on defence and political rights on the basis of a cross-country data set.

Keywords: Automated modelling, genetic algorithm, Bayesian information criterion.

1 Introduction

Automated modelling is a recent phenomenon in empirical social sciences, i.e. in economics, sociology and political science. Similar to the phenomenon of the increasing availability of data and variables, automated modelling is worth being considered for applied empirical problems, be it the explanation of variables by other variables or the forecasting of variables. For an overview of different approaches in automated modelling see also [7]. However, social scientists prefer models that are based on human theoretical reasoning rather than on models that are based on automated (data-driven) approaches. While there is, of course, much in favour of human reasoning and model building one problem is that theory in social science is usually not precise enough in giving exact information on which set of variables to use for explaining true and important empirical relationships. This is especially true when there are a number of alternative variables available that explain nearly the same phenomenon. When there are enough observations available we could use all the available variables and the data will show which variables are important. However, usually the number of observations is quite limited. For example, when a simple linear regression model is estimated on the basis of a large set of variables, we soon discover that the parameter estimates obtained for a lot of variables are not significantly different from zero.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 912 – 919, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Given this situation we might conclude that the model is not appropriate since a lot of explanatory variables are not able to explain the variable of interest (dependent variable) and the model size is too large. The set of insignificant variables may obliterate the quality of estimates. A selection might help. But which variables should remain in the model and which variables should be excluded? This question is of special concern when independent variables are (highly) correlated. However, there is uncertainty about which subset of variables belong in the model and which variables are unimportant. Automated modelling approaches are able to give an answer to this question.

The aim of this paper is to present the use of a genetic algorithm (GA) as a method for automated modelling in social sciences since in the recent debate on algorithms for analysing causal structure among variables genetic algorithms have rarely been applied. This paper argues that automated modelling and model selection using heuristic algorithms is favourable compared to frequently applied approaches in social sciences such as Bayesian Model Averaging (BMA). It is shown that genetic algorithms used for Bayesian Model Selection (BMS) are able to identify parsimonious and robust causal structures in a reasonably short time. The advantages of using a genetic algorithm as the method for automated modelling is exemplified by the identification of previously unknown but important causal relationships for long-term inflation, defence spending and political rights on the basis of a cross-country data set.

The structure of the paper is as follows: In chapter 2 the problem of model and variable uncertainty in social sciences is outlined. In chapter 3 the use of a GA for the problem of model selection is presented and a GA is applied for the identification of models for three different variables. Also, the empirical results are discussed with respect to the structure of the identified models. In chapter 4 the efficiency of the GA is critically investigated and discussed, and in chapter 5 conclusions are drawn and a summary made.

2 Model Selection, Uncertainty and Information Criteria

A model is a representation of real-world phenomena. Usually, setting up a model involves choices regarding what to include and what to exclude. In social science, the structure of a model is often thought of in terms of equations used to describe relationships among variables. Social scientists are usually interested in the interpretation of models and favour simple models as they are easy to interpret. Also, in social science linear relationships are popular. Compared to non-linear relationships, linear relationships are also easier to interpret. Considering the necessity to select models that are straightforward to interpret, this paper considers linear (regression) models. The focus on linear regression models reduces the selection exercise at hand to a variable selection problem. However, in the presence of a large number of variables that are potentially relevant for the explanation of another variable, a selection has to be made for statistical reasons, i.e. high correlations among variables as well as for reasons of interpretation, i.e. parsimonious solutions are preferable. But which variables should be selected? Which variables are relevant and which are not?

There are two categories of strategies that are frequently applied in empirical social sciences for the systematic identification of relevant empirical relationships, i.e. rele-

vant sets of variables. One is based on the idea of averaging over alternatives and the other is based on selecting an alternative. For an overview see [5] and [4]. The question of selection and averaging arises when the relationship between *y* (dependent variable) and a subset of *x*'s (potentially informative determinants) has to be modelled, but there is uncertainty about which subset to use. The fundamental difference in the idea behind both strategies is that BMS assumes the existence of a true model while BMA assumes that there is no true model that describes the data generating process. In this sense, BMS selects one out of many alternative models – models not selected are not used – while BMA keeps all models, each assigned a weight. Model selection refers to using the *data* to select one model from many possible models M_1, …, M_r, while model averaging refers to the process of estimating some quantity under each model M_j and then averaging the estimates according to how likely each model is. In general, according to [10] a model can be interpreted as a set of probability distributions. Suppose that data consist of a normally distributed outcome *y* and a covariate *x* and suppose that two possibilities are entertained. Possibility one is that *y* is unrelated to *x* and the second possibility is that *y* is linearly related to *x*. Then M_1 consists of the distributions for which $y \sim N(\mu, \sigma^2)$ and M_2 consists of the distributions for which $y \sim N(\beta_0 + \beta_1 x, \tau^2)$. When more models are considered, model M_j is used together with the data to predict $\hat{y}_j$. This implies an overall prediction $\sum_{j=1}^{r} w_j \hat{y}_j$ where a particular weight w_j is used to express the probability that the model M_j generated the data. The BMA way is to compute the posterior probability for each model $P(M_j \mid data)$. In the case of model selection, a model that maximises $P(M_j \mid data)$ is chosen. For a detailed explanation of the derivation of $P(M_j \mid data)$ and the strengths and limitations see [10] and [3]. However, BMA is currently very popular in empirical social science, although there are some disadvantages of BMA. BMA is not able to choose between variables that are (highly) collinear and are explaining (nearly) the same phenomenon. Another problem of BMA is that its application will usually not result in a parsimonious model. A last problem is that, when there is a 'true' data generating process, then the results of BMA are biased as 'bad' models are used for inference. Independent of the latter argument we assume the first two arguments against BMA to be strong enough for choosing model selection to solve the problem at hand. In general, model selection is the task of selecting a model (set of variables) from a set of potential models. Model selection is a bias versus variance trade-off and relates to the statistical principle of parsimony. Inference using models with too few variables can be biased. On the other hand, models with too many variables may be poor in their precision or identify effects that are spurious. As an operationalisation to balance the bias versus variance trade-off, statistical literature offers a number of information criteria. For the problem at hand we decide to use the well-known and frequently applied Bayesian Information criterion (BIC), see [9], as it is relatively strict in the identification of relevant variables, or in the identification of a relevant set of variables, respectively. The idea of the BIC is to penalise the log-likelihood by $2nf(T)/T$ for *k* parameters and a sample size of *T*, where: $BIC = \ln \tilde{\sigma}^2 + n \ln(T)/T$.

The penalty term reflects the costs of over-parameterisation. However, a disadvantage of model selection is that, in the presence of a high number of variables (k is the total number of variables), finding the optimal BIC model is challenging because there are 2^k possible solutions. For the problem at hand it is not feasible to evaluate all possible specifications. To overcome this problem we use the possibilities of a GA to find the optimal BIC solution in reasonable time and with reasonable computational effort. The problem of using GA is that it does not provide a guarantee that the optimal solution will be found. By making robustness checks on the GA we will investigate the efficiency of the algorithm.

3 Data and Empirical Results

On the basis of a cross-country data set containing 67 explanatory variables (including economic, sociological, political and structural variables) a GA is applied to find the set of variables that best explains three different variables in a linear regression

Table 1. Specifications identified by the genetic algorithm

Average Inflation 1960-90		Defence Spending Share		Political Rights	
C	-38,0963**	C	-0,0119***	C	8,2592***
	(18,39764)		(0,0034)		(0,5739)
Distance to Big Cities	0,0020***	Public Education Spending Share	-0,9530***	Civilian Liberties	-2,9858***
	(0,0005)		(0,0371)		(0,3460)
Fraction Buddhist	-27,3403***	Government Share of GDP	0,9887***	Pop. Density Coastal	0,0004**
	(7,2532)		(0,0132)		(0,0002)
European dummy	-30,7737***	Gov. Consumption Share	-0,9855***	European Dummy	-2,2914***
	(5,3030)		(0,0135)		(0,4846)
Fertility in 1960s	-32,9508***	Life Expectancy	0,0002***	Fraction Hindus	-2,7004***
	(5,6373)		(4,7E-05)		(0,6826)
Religion Measure	21,7631***	Fraction Muslim	0,0053***	Life Expectancy	-0,0547***
	(5,9305)		(0,0011)		(0,0125)
Primary Exports	17,7670***	Political Rights	0,0005**	Area Near Navigable Water	-0,9180***
	(6,1421)		(0,0002)		(0,2798)
Revoluti -ons	16,7791***	Exchange Rate Distortions	-2,6E-05***	Fraction GDP in Mining	-2,7695**
	(4,9918)		(8,6E-06)		(1,1542)
Size of Economy	4,26313***	Revolutions and Coups	0,0032**	Population Over 65	22,5588***
	(0,7902)		(0,0013)		(6,9444)
		Share Population in Tropics	0,0046***		
			(0,0012)		
Variables	8	Variables	9	Variables	8
R^2	0,6192	R^2	0,9900	R^2	0,8645
Adj. R^2	0,5806	Adj. R^2	0,9888	Adj. R^2	0,8508
BIC	7,7337	BIC	-8,6767	BIC	2,6684
N	88	N	88	N	88
F(Prob.)	0.0000	F(Prob.)	0.0000	F(Prob.)	0.0000

Note: Estimations according to OLS. Numbers in parenthesis indicate standard errors of coefficients. Significance: *** $\alpha \leq 0.01$, ** $\alpha \leq 0.05$ and * $\alpha \leq 0.1$. For details on the variables see [8]. Please note that higher values of the variable political rights imply less political rights.

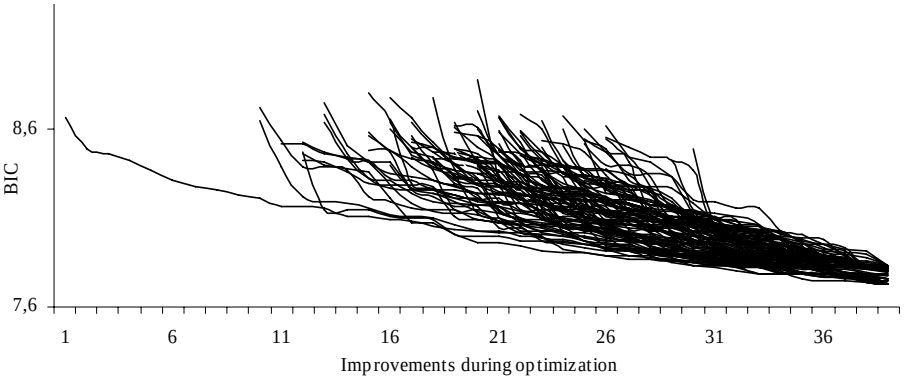
model according to the BIC. The fitness function (goal function) of the GA is defined by the BIC. The cross-country data set considers 88 countries and the variables are usually averages of the years 1960 to 1995. This means that long-run relationships are expressed in data. For details on the data set see [8]. A consideration of 67 potentially important variables results in 2^{67} possible solutions. By assuming that calculation of one solution takes one nanosecond, complete calculation of every solution would take 4680 years. We expect that the utilisation of GA solves the problem in a reasonable time. We apply the GA to three dependent variables: Average inflation rate between 1960-1990 according to [6], average share of public expenditures on defence as a fraction of GDP between 1960 and 1965 according to [2] and the political rights index from [1]. The models that are identified by the GA can be seen in Table 1.

As can be seen the automated model selection approach on the basis of a GA resulted for each of the three dependent variables in a parsimonious solution, i.e. 8 to 9 variables are in the models. Every specification is characterised by highly significant estimates and a high explanation of total variance, which raises confidence in the results remarkably. By looking at the variables that are identified to be important, we can see that variables concerning religion are frequently included. For example, the fraction of Buddhists is found to be important in describing long-term average inflation, the fraction of Muslims is related to defence spending and the fraction of Hindus is related to political rights. However, the frequent 'appearance' of religious variables in explaining long-term phenomena in economics, sociology and political science needs a more detailed discussion but goes along with other results of automated modelling, see[8]. For reasons of space a detailed discussion of the models is not included here, in particular the question whether variables in the models are spurious or not.

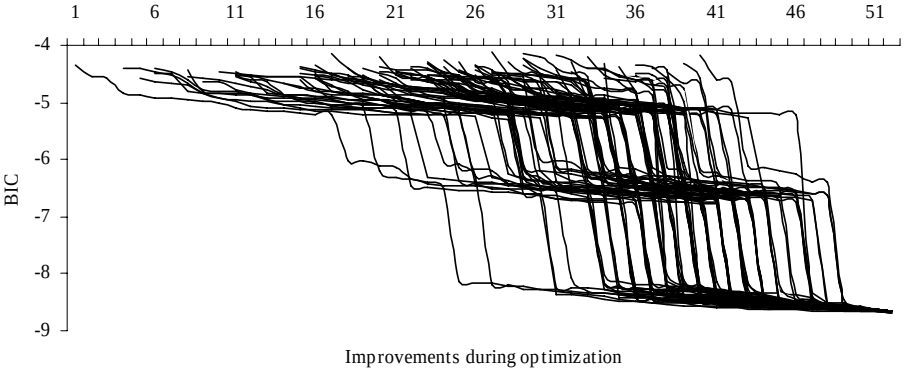
4 The Efficiency of the Genetic Algorithm

Now a closer look at the efficiency of the GA in finding the optimal BIC solution is made. By thinking about the size of the optimisation problem (i.e. there are 2^{67} possible solutions) it is very likely that there are a lot of local optima. To investigate how the GA performed in finding the global optimum the search is repeated for each dependent variable 100 times, i.e. 100 runs are performed. If the application of the GA (nearly) always results in the same solution, and this is the best solution found, we can be very confident that indeed the optimal solution was found. In Fig. 1 the results of the 100 runs are plotted for each of the three variables of interest.

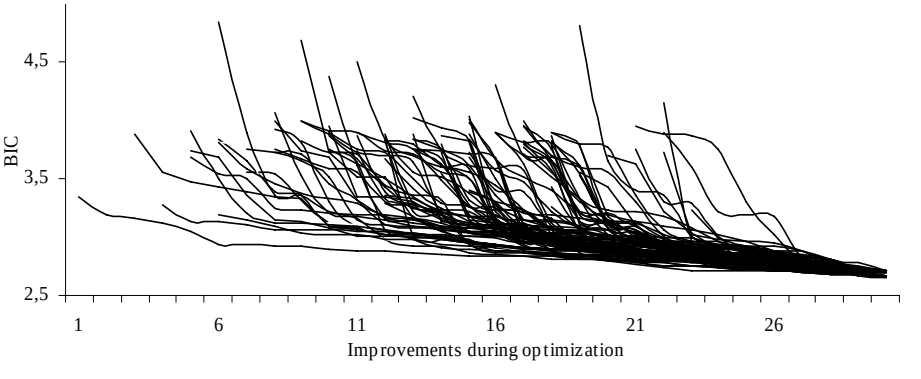
It can be seen in Fig. 1 that during the optimisation process the GA improves substantially the BIC-quality of the specifications. Each graph in Fig. 1 represents improvements during the optimisation process. As the BIC is due to minimisation we can see that the starting solution (which is randomly chosen) is far worse than the solution to which the GA converged. But has every run (for the respective dependent variable) converged to the same solution? We can see that for average inflation (Fig. 1 (a)) convergence was only achievable to an area. This means that only 20 % of the runs converged to the optimal solution (that was found). Unfortunately 30 % of the runs converged to a solution of a higher BIC, which was the solution most frequently found – but which is, unfortunately, definitely not the optimal solution. In the case of defence spending, see Fig. 1 (b), we can see that convergence is closer. 91 % of all



(a) Inflation



(b) Defence spending



(c) Political rights

Notes: The robustness of convergence to the optimal specification was investigated by 100 runs. The graphs indicate improvements during optimisation. Graphs have different lengths because of different (randomly) chosen starting points, i.e. the shorter the graphs, the faster convergence and vice versa.

Fig. 1. Robustness of convergence to the global optimum of the genetic algorithms

runs converged to a solution. Again, unfortunately this solution is not the optimal solution because 4 % of all runs resulted in the best found solution, which is very close to the solution which has been found most frequently. The same picture can be seen for political rights, see Fig. 1 (c). Again, convergence of all runs is very similar and the region of convergence is very close but, unfortunately, the solution most frequently found (73 %) is not the optimal solution. The best solution found was only reached by 18 % of the runs. Nevertheless, using a GA enables the identification of solutions with high explanatory capacity and with high significance. Even though the identification of the best solution is not robust and many runs are needed to prevent getting stuck in local minima. However, even though there is heterogeneity of convergence it has to be noted that this heterogeneity is small when looking at the variables in the models. This means that variables of the optimal solution and the solution that was most frequently found only differ slightly, i.e. only one or two variables are differing. Nevertheless, the heterogeneity in convergence gives rise for further research. Especially fruitful would be the combination of a GA with local search algorithms. This means that the GA is able to find the area near the global optimum, but has a problem in exactly reaching the global optimum. Therefore, the stochastic search process can be made more efficient by using local search (neighbourhood search) algorithms at the end of the GA search process.

5 Conclusions

This paper argued that there is uncertainty in social science, i.e. economics, sociology and political science, about empirical relationships and optimal specifications for explaining social science phenomena. This uncertainty is triggered by the phenomenon that the number of variables that are available for empirical models is rapidly increasing. Model selection with information criteria offers a solution to the reduction of model uncertainty. The problem of model selection in the presence of a large number of variables that are potentially important is that the total number of possible solutions is extremely large and computationally demanding, i.e. it is not feasible to calculate and evaluate all possible solutions in a reasonable period of time. This paper faced the problem of variable selection for linear regressions as a discrete (global) optimisation problem and argued that stochastic optimisation or search algorithms can be applied to solve the optimisation problem in reasonable time. GAs are stochastic search algorithms and perform a systematic search for the optimal set of variables (the optimal specification) automatically or data-driven, respectively. As the characteristics of the presented approach are that the search for an optimal model is systematic and data-driven, i.e. automated, the approach can be classified as an automated modelling approach.

To exemplify the efficiency and usefulness of genetic algorithms in this work a GA was applied for the identification of long-run inflation, defence spending and political rights on the basis of a cross-country data set containing 67 potentially informative variables. The results of the exercise showed that model selection using a GA resulted for all three dependent variables in the identification of specifications that are: (i) parsimonious (which has the advantage that the results and the models are easy to interpret), (ii) characterised by a high estimation quality (i.e. the significance of all

variables is very high which gives confidence in the explanation), (iii) able to explain a lot of the total variance (i.e. have a high R^2 and therefore are able to explain the phenomenon) and (iv) interesting with respect to theoretical reasoning.

References

1. Barro, R.J.: Determinants of democracy. Journal of Political Economy 107, 158–183 (1999)
2. Barro, R.J., Lee, J.-W.: International comparisons of educational attainment. Journal of Monetary Economics 32, 363–394 (1993)
3. Clyde, M.: Bayesian Model Averaging and Model Search Strategies. Bayesian Statistics 6, 157–185 (1999)
4. George, E.I., McCulloch, R.E.: Approaches for Bayesian variable selection. Statistica Sinica 7, 339–373 (1997)
5. Hoeting, J.A., Madigan, D., Raferty, A.E., Volinsky, C.T.: Bayesian Model Averaging - A Tutorial. Statistical Science 14, 382–417 (1999)
6. Levine, R.E., Renelt, D.: A sensitivity analysis of cross-country growth regressions. American Economic Review 82, 942–963 (1992)
7. Phillips, P.C.B.: Automated Discovery in Econometrics. Econometric Theory 21, 3–20 (2005)
8. Sala-i-Martin, X., Doppelhofer, G., Miller, R.I.: Determinants of Long-Term Growth - A Bayesian Averaging of Classical Estimates (BACE) Approach. American Economic Review 94, 813–835 (2004)
9. Schwarz, G.: Estimating the dimensions of a model. Annals of Statistics 6, 461–464 (1978)
10. Wasserman, L.: Bayesian Model Selection and Model Averaging. Paper presented at the Mathematical Psychology Symposium on Methods for Model selection 1997 in Bloomington, Indiana (1997)

Extending the Growing Neural Gas Classifier for Context Recognition

Rene Mayrhofer[1] and Harald Radi[2]

[1] Lancaster University, Infolab21, South Drive, Lancaster, LA1 4WA, UK
rene@comp.lancs.ac.uk
[2] Tumpenweg 528, 5084 Grossgmain, AT
harald.radi@nme.at

Abstract. Context awareness is one of the building blocks of many applications in pervasive computing. Recognizing the current context of a user or device, that is, the situation in which some action happens, often requires dealing with data from different sensors, and thus different domains. The Growing Neural Gas algorithm is a classification algorithm especially designed for un-supervised learning of unknown input distributions; a variation, the Lifelong Growing Neural Gas (LLGNG), is well suited for arbitrary long periods of learning, as its internal parameters are self-adaptive. These features are ideal for automatically classifying sensor data to recognize user or device context. However, as most classification algorithms, in its standard form it is only suitable for numerical input data. Many sensors which are available on current information appliances are nominal or ordinal in type, making their use difficult. Additionally, the automatically created clusters are usually too fine-grained to distinguish user-context on an application level. This paper presents general and heuristic extensions to the LLGNG classifier which allow its direct application for context recognition. On a real-world data set with two months of heterogeneous data from different sensors, the extended LLGNG classifier compares favorably to k-means and SOM classifiers.

1 Introduction

Context Awareness, as a research topic, is concerned with the environment a user or device is situated in. Although its roots probably lie in robotics [1], the advantages of making applications in the fields of pervasive, ubiquitous or mobile computing aware of the user or device context are obvious: user interaction and application behavior can be adapted to the current situation, making devices and their applications easier to use and making more efficient use of the device resources.

Our approach to making devices context aware is based on three steps: sensor data acquisition, feature extraction and classification [2]. In these three steps, high level context information is inferred from low level sensor data. There are two key issues in this approach: the significance of acquired sensor data, and the use of domain-specific heuristics to extract appropriate features. A broad view on

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 920–927, 2007.
© Springer-Verlag Berlin Heidelberg 2007

the current context is necessary for many applications, and can only be derived with a multitude of sensors with ideally orthogonal views of the environment. Using multiple simple sensors and merging their data at the classification step allows to use different heuristics for each sensor domain, and thus profit from well-known methods in the specific areas.

Examples for simple sensors that are relevant to typical context-aware applications are Bluetooth or wireless LAN adapters, which can capture parts of the network aspect, the GSM cell ID, which defines a qualitative location aspect, or the name of the currently active application (or application part), which captures a part of the activity aspect of a user or device context. Other, more traditional sensors include microphones and accelerometers to capture the activity aspect [3], video cameras, or simple light and temperature sensors. The types of values provided by these sensors are very different. A method to cope with this heterogeneity of features has first been presented in [4] for the general case, independent of the used classification algorithm. This paper is concerned with necessary modifications to the Growing Neural Gas (*GNG*) classifier.

GNG has been introduced by Bernd Fritzke in 1995 [5] and shares a number of properties with the conceptually similar Dynamic Cell Structure algorithm, which has been developed independently by Jörg Bruske and Gerald Sommer [6]. In 1998, Fred Hamker extended GNG to support life-long learning, addressing the Stability-Plasticity Dilemma [7]; the resulting algorithm has also been called Lifelong Growing Neural Gas (*LLGNG*). The main difference between GNG and LLGNG is that the latter uses local error counters at each node to prevent unbounded insertion of new nodes, thus allowing on-line learning for arbitrary long periods.

The basic principle of GNG and LLGNG is to insert new nodes (clusters) based on local error, i.e. it inserts nodes where the input distribution is not well represented by clusters. For classification of features to recognize context, this is an important property because it ensures independence of the usually unknown input distribution. Additionally, edges are created between the "winner" node, which is closest to the presented sample (in this case a feature vector) and the second nearest one; other edges of the winner node are aged and removed after some maximum age. The resulting data structure is a cyclic, undirected, weighted graph of nodes and edges, which is continuously updated by competitive Hebbian learning. In LLGNG, nodes and edges are created and removed based on local criteria. For details on the insertion and removal criteria, we refer to [7].

2 Extensions to LLGNG

For using LLGNG for context recognition, we extend it in two areas:

2.1 Extension 1: Coping with Heterogeneous Features

To ease the implementation, we reduce the set of heterogeneous features to a minimal subset of abstract features providing meaningful implementations of the necessary two operations *getDistance* and *moveTowards* (see also [4]).

Implementations of these methods may be general for certain types of features, but will typically benefit from domain-specific heuristics. Knowledge about the respective sensing technology should be applied in the implementation of these two methods.

For a framework for context awareness and prediction [2] on mobile device, we found that many of the typical features can be reduced to a common set of basic features. Currently, we use base classes *AbstractString*, *AbstractStringList*, *NumericalContinuous*, *NumericalDiscrete*, and *Binary* for the actual implementations. The *NumericalContinous*, *NumericalDiscrete*, and *Binary* features implement *getDistance* and *moveTowards* as the Euclidean distance metric and thus need no special considerations.

On the other hand, the *AbstractString* feature serves as a base class for features that return a single-dimensional output value that can be represented as a string (e.g. WLAN SSID, MAC address, GSM cell ID, etc.). Although not the best solution, using the string representation as similarity measure for the feature values is still more meaningful than having no metric at all. For *getDistance*, we defined the Levenshtein distance (normalized to the longest encountered string) as distance metric. For *moveTowards*, our extended version of the Levenshtein algorithm also applies these operations with a given probability and returns a string that is somewhere between (in terms of our distance metric) the compared strings and represents the actual cluster position for this feature dimension.

The *AbstractStringList* feature serves as a base class for features that return a set of output values per sample point (e.g. list of peers, list of connected devices, etc.) that cannot be easily represented by a single string. Based on the idea of using the Levenshtein distance, we assign each list element a unique index and compose a bit vector from the string list. If a given string is present in the list, its corresponding bit in the bit vector is set. The distance metric for *getDistance* is then defined as the Hamming distance between the bit vectors of two given lists. Again, the *moveTowards* function can compose an intermediate bit vector that represents the actual cluster position.

Our extended version of LLGNG simply uses *getDistance* and *moveTowards* on each dimension instead of applying Euclidean metrics.

2.2 Extension 2: Meta Clusters

In the standard formulation of GNG and LLGNG, the edges in the internal graph are only used for three purposes:

- for adapting neighbors (adjacent nodes) of the winner
- for inserting a new node between the winner and its neighbor
- for removing nodes which have lost all edges due to edge aging

Additionally, the local insertion and removal criteria in LLGNG depend on the neighbors. As can be seen, the edges are not used for analyzing the graph structure itself.

One of the problems with using standard, unsupervised clustering algorithms for context recognition is that the automatically created clusters are usually too

fine-grained for mapping them to high-level context information like "in a meeting" or "at home"; for defining simple context based rules on the application level, we would like to achieve this granularity. Therefore, we introduce the concept of meta clusters by explicitly using properties of the generated graph structure. This can be seen as a heuristic that is independent of the problem domain, but that depends on the internal (LL)GNG graph.

The (LL)GNG graph, after some learning time, usually consists of multiple components distributed over the cluster space. These components consist of two or more connected nodes and are perfect candidates for high-level context information, because they cover arbitrarily shaped areas in the high-dimensional, heterogeneous cluster space instead of only RBF-type shapes that single clusters cover. In our extended version, we assign each component a unique meta cluster ID and use this ID for mapping to high level context information. For performance reasons, the meta cluster IDs can not be recalculated after each step, but have to be updated during on-line learning. When starting with two adjacent nodes in the initialization phase, we simply assign the first meta cluster ID to this component and cache the ID in each node. During on-line learning, insertion and removal of edges will lead to the following cases:

- inserting a new node: Since a new node will only be inserted between two existing ones, its meta cluster ID is set to the ID of the connected nodes.
- inserting an edge between two nodes with the same ID: No change is necessary.
- inserting an edge between two nodes with different ID: Due to *merging* two components, one of the IDs will be used for the resulting component, overwriting the other one. When both IDs have already been assigned to high level context information, the merge is prevented by not inserting the edge.
- removing an edge: If the previously directly adjacent nodes are no longer connected via another path, a meta cluster *split* has occured and a new meta cluster ID must be allocated for one of the two components.

Normal adaptation of clusters has no influence on the graph structure and can thus be ignored for the handling of meta clusters. Further (performance) optimizations for meta cluster handling in our extension include a caching of meta cluster IDs and an incremental check if two nodes are still connected after removing an edge.

2.3 Performance Optimizations

Finding the nearest and second nearest cluster to a sample vector is a common task performed by (LL)GNG. Our chosen architecture allows different types of features in every dimension of the cluster space; therefore, comparisons have to be performed separately for every dimension and cannot be optimized. One option is to limit the amount of necessary comparisons to an absolute minimum. To accomplish this, we store every cluster in a splaytree sorted by their distance from the origin. A splaytree has the advantage that recently accessed nodes are kept on the very top of the tree and infrequently used nodes move towards the

bottom. Assuming that samples aren't evenly distributed in the cluster space but accumulated, the nodes of interest are always on the top of the tree and the tree only has to be traversed completely if the sample data changes spontaneously.

Additional optimizations were done by caching results of internal computations for the insertion and removal criteria as defined in [7]. These helper variables are only recomputed when the node position is changed instead of each time a neighbor is queried. More details on these optimizations are presented in [2,8].

3 Evaluation

Our extended LLGNG algorithm has been compared to the more well-known classification algorithms k-means and Kohonen Self-Organizing Map (SOM) both with artificial and real-world data sets. Details on the comparison can be found in [2].

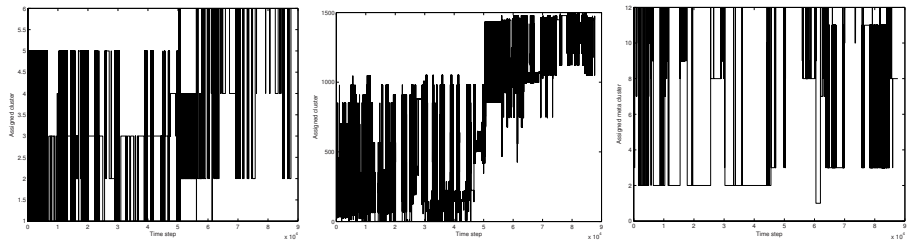
3.1 Data Set

The real-world data set used in this evaluation has been gathered continuously over a period of about two months on a standard notebook computer which was used for daily work. No special considerations were taken during the use of the notebook regarding context recognition. Therefore, the data set should provide representative sensor data for the chosen scenario. A wide range of sensors was used, including a microphone, the active/foreground window, if it was plugged into its charger, WLAN, and GSM. Domain-specific heuristics for these sensors are used to extract 28 different features. At one sample every 30 seconds, roughly 90000 samples of 28 dimensions were collected.

3.2 Pre-processing and Classification Error

Classification error was defined as the average distance between each data point and its respective best matching cluster after training has been completed, which is similar to the cost function minimized by k-means. This is a universal criterion suitable for evaluating the classification quality of arbitrary unsupervised clustering algorithms, and it is already well-defined and used in different implementations like the SOM Toolbox for Matlab; lower values for the classification error represent a better adaptation to the feature values and thus a more accurate classification.

Both k-means and SOM are used in a batch training mode and are thus not susceptible to initial transients, while LLGNG suffers from such effects due to its online mode. For k-means and SOM, the data set has been pre-processed to transform all non-numerical into numerical (binary) input dimensions by assigning one input dimension for each possible feature value, i.e. the one-of-C method. This transformation yields a 198 dimensional input space with a large number of binary inputs. All dimensions are further normalized to $[0; 1]$, as recommended by standard literature on clustering. Since the implementations of the distance



(a) K-means (b) SOM (c) Extended LLGNG

Fig. 1. Cluster trajectories computed by the different classifiers

metrics specific to each feature are also normalized to $[0; 1]$, the overall classification error is assumed to be comparable even if the number of input dimensions is different.

3.3 K-Means

K-means clustering divides a given data set into k clusters by minimizing the sum of distances between the data points and cluster centers, whereas k has to be predetermined. Thus, k-means is actually not usable for live context classification, and is only used for comparison. The k-means clustering implementation in the Statistics Toolbox for Matlab was used iteratively for $k = 2 \ldots 40$ to determine the optimal number of clusters. With the optimum of 6 clusters, k-means reached a final classification error of 0.7451.

In Fig. 1a, the assigned clusters are depicted for each time step in the initial feature data set. The trajectory seems unstable and oscillates quickly between contexts. K-means clustering in this form is infeasible for embedded systems due to the enormous computational power necessary to optimize the number of clusters; determining the number of clusters for this test case took over 2 hours with 5 computers similar to our reference machine being used in parallel.

3.4 Kohonen Self-organizing Map

The Kohonen SOM is, in contrast to k-means, a soft clustering approach: each input sample is assigned to each cluster with a certain degree of membership. For the purpose of this comparison, this can easily be reduced to hard clustering by searching for the so-called "best matching unit", which is then assumed to be the context class for the respective time step. The following evaluations were performed with the specialized SOM Toolbox for Matlab developed by the Laboratory of Computer and Information Science at Helsinki University of Technology because of its flexibility and simple means of visualization.

Training a new SOM with the whole data set took 690 seconds and results in a final classification error of 0.5659. The SOM grid is automatically set to 71x21 clusters with a heuristic embedded in the toolbox. The u-matrix indicates around 4 to 8 larger areas, which seems reasonable when compared to the 6

clusters found by the k-means method and which can be seen as some form of meta clusters. Although the final classification error is lower than for k-means clustering, this is not surprising because of the significantly larger number of cluster prototypes that are available for mapping the input space. The large number of clusters formed by the SOM could not be used for directly representing high-level context, but a second classification step would be necessary.

The clusters assigned to the feature values for each time step are shown in Fig. 1b as the numbers of respective best matching units. The trajectory also shows oscillating behavior. Without a second step of clustering to exploit some form of meta clusters formed by the SOM, the resulting cluster trajectories seem unusable for context prediction on an abstract level. Additionally, the trajectory of the whole data set presented in Fig. 1b shows signs of unfavorable separation of clusters into areas in the input space around time step 50000: the apparent switch to a completely separate region of cluster prototypes is not supported by the visualization of the feature values. A change to different clusters around the time step 50000 is also visible in the k-means trajectory in Fig. 1a, but it is not as drastic as for the SOM.

3.5 Extended LLGNG

Prior to training the extended LLGNG algorithm with this test data set, a simulated annealing procedure was used to optimize some of its static parameters. However, the optimization does not significantly improve the classification error, suggesting stability against changes in the initial conditions or parameter sets and supporting the findings reported in [9].

One-pass training with the whole data set took only 474 seconds, produces 103 clusters composing 9 meta clusters and achieves a final classification error of only 0.0069. This means that the input data distribution is well approximated by the automatically found clusters, even with the inherent noise in our real-world data set and significantly less clusters than used by the SOM ($71 \cdot 21 = 1491$).

Fig. 1c shows the best matching meta clusters for each time step; the meta clusters are a higher-level concept than clusters and are thus suited better as abstract context identifiers. When comparing the trajectories computed by k-means, SOM, and the extended LLGNG variant, the latter one is more stable and shows fewer oscillations, with the notable exception of the time frame between time steps 64000 and 70000 where the LLGNG trajectory also shows oscillating behavior.

4 Conclusions

Lifelong Growing Neural Gas, a variant of the Growing Neural Gas classification algorithm, is an ideal algorithm for context recognition because it is optimized towards continuously running, un-supervised classification. In this paper, we presented necessary extensions for applying it to heterogeneous input data and for directly assigned high level context information to its output as well as performance optimizations. While k-means and the Kohonen Self-Organizing

Map (SOM) produced classification errors of 0.7451 and 0.5659, respectively, our extended LLGNG classifier achieved an error of only 0.0069.

It should be noted that, unlike SOM and k-means, the extended LLGNG is used in online mode, which is far more challenging. Without any further changes, the algorithm can immediately be used for continuous learning over arbitrary periods of time. K-means and SOM both had to be trained to produce the above results, with each sample being presented numerous times for training. The extended LLGNG only had a single pass over the whole data set and still achieves a significantly lower classification error and more stable results. One suspected reason for this success is that our extensions that enable LLGNG to deal directly with heterogeneous data indeed lead to higher classification quality due to the lower-dimensional input space and due to the preservation of the semantics of all feature dimensions.

References

1. Rosenstein, M.T., Cohen, P.R.: Continuous categories for a mobile robot. In: Proc. AAAI/IAAI: 16th National/11th Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence, pp. 634–640 (1999)
2. Mayrhofer, R.: An Architecture for Context Prediction. PhD thesis, Johannes Kepler University of Linz, Austria (October 2004)
3. Lukowicz, P., Ward, J.A., Junker, H., Stäger, M., Tröester, G., Atrash, A., Starner, T.: Recognizing workshop activity using body worn microphones and accelerometers. In: Ferscha, A., Mattern, F. (eds.) PERVASIVE 2004. LNCS, vol. 3001, pp. 18–32. Springer, Heidelberg (2004)
4. Mayrhofer, R., Radi, H., Ferscha, A.: Feature extraction in wireless personal and local area networks. In: Proc. MWCN 2003: 5th International Conference on Mobile and Wireless Communications Networks, October 2003, pp. 195–198. World Scientific, Singapore (2003)
5. Fritzke, B.: A growing neural gas network learns topologies. In: Advances in Neural Information Processing Systems, vol. 7, pp. 625–632. MIT Press, Cambridge MA (1995)
6. Bruske, J., Sommer, G.: Dynamic cell structure learns perfectly topology preserving map. Neural Computation 7, 845–865 (1995)
7. Hamker, F.H.: Life-long learning cell structures—continuously learning without catastrophic interference. Neural Networks 14(4–5), 551–573 (2001)
8. Radi, H.: Adding smartness to mobile devices - recognizing context by learning from user habits. Master's thesis, Johannes Kepler University Linz, Austria (2005)
9. Daszykowski, M., Walczak, B., Massart, D.L.: On the optimal partitioning of data with k-means, growing k-means, neural gas, and growing neural gas. Journal of Chemical Information and Computer Sciences 42(1), 1378–1389 (2002)

Neural Network Based Path Detection for an FMCW Positioning System

Ralf Mosshammer[1], Florian Frank[1], and Mario Huemer[2]

[1] Institute for Electronics Engineering,
University of Erlangen-Nuremberg,
Cauerstrasse 9, 91058 Erlangen, Germany
[2] Chair of Embedded Systems, NES
Klagenfurt University
Lakeside B02b, 9020 Klagenfurt, Austria
{mosshammer, frank}@lfte.de, mario.huemer@uni-klu.ac.at

Abstract. Multipath propagation is a major source of error for runtime detection based positioning systems. For the case of an FMCW-based positioning system, the overlap of pulse shapes in the frequency domain restricts exact measurement of frequency, and thus of runtime. Choosing a measurement point on the slope of the pulse is a way to mitigate measurement errors. In this paper, we present a neural network as a means of estimating the ideal measurement point. The network is shown to outperform fixed level measurements even with very sparse training data.

Keywords. FMCW, Positioning, Artificial Neural Network, Path Detection.

1 Introduction

Microwave ranging and positioning techniques have emerged as a dominant industry branch in recent years [1]. This development has mainly been driven by two distinct application fields: consumer outdoor positioning applications, such as the popular Global Positioning System (GPS) or cellphone network based ranging methods; and industrial and logistics applications, which are in general confined to a specific area, such as a factory hall, and mostly indoors.

In the latter field, customer interest lies mainly in saving costs through intelligent, location-aware materials transportation and storage. The specific demands for an industrial positioning system are distinctly different from person localization: for large amounts of processed items, the system must support a high number of concurrent clients, feature real-time update rates and very high accuracy and resolution.

Within the EU funded RESOLUTION project, a number of partners from the industrial and academic world are working on an integrated system solution for wireless microwave positioning [2]. The envisioned target markets for this system are the aforementioned logistics applications as well as indoor personal positioning. To accommodate these, the system will feature an integrated communications module based on the IEEE 802.11 wireless LAN standard.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 928–935, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The major challenges for the RESOLUTION positioning chip are the high update rate, which is required for robotics, the accommodation of a potentially high number of clients, and the positioning accuracy. For the latter parameter, a target value of 5 cm was specified by market analysis. This value must be achieved despite the spectral limitations of the chosen ISM band at 5.8 GHz.

In this work, we present a method for adapting the measurement process in baseband by means of a neural network (NN). Extensive simulation results corroborate our claim that this method improves positioning accuracy in the presence of multipath, with manageable implementation effort.

This paper is organized as follows. Section 2 gives an overview of the signal processing properties of the RESOLUTION system and introduces the indoor multipath propagation problem. Section 3 outlines the basics of neural network design and training algorithms. Section 4 proves our method by giving comprehensive simulation results, followed by the summary.

2 System Overview

The RESOLUTION positioning system is based on frequency modulated continuous wave (FMCW) radar, a principle which has been shown to exhibit excellent characteristics in earlier works [3]. The RESOLUTION system utilizes a bistatic radar approach, which means that transmitter and receiver are two separate hardware entities, i.e. a signal is transmitted and received by different antennas.

In principle, the system operates in a round trip time of flight (RTOF) protocol, where the actual position calculation takes place in the station initiating the positioning process, as visualized in Fig. 1. Take note that instead of simply reflecting the incident signal from the transmitter, station B actually generates a ramp of its own. This means that signal power loss is only proportional to d^2 instead of d^4, where d is the line of sight distance between station A and B.

In the following, we give a system theoretical description of a ranging operation involving one transmitting and one receiving unit. We start by writing the ideal transmit signal as

$$s_{\mathrm{TX}}(t) = \cos\big((\mu t)t\big) \ , \tag{1}$$

where μ is the slope of the frequency ramp and is given by the ratio of the system bandwidth B to the ramp period T. In the 5.8 GHz ISM band, spectral emission is regulated to a maximum of 150 MHz bandwidth, which limits the achievable accuracy in the presence of multipath propagation.

The ideal FMCW signal is transmitted over a noisy multipath channel, resulting in a received signal of the general form

$$s_{\mathrm{RX}}(t) = \sum_{i=1}^{N_c} \alpha_i s_{\mathrm{TX}}\big(t - \tau_i\big) + n(t) \ . \tag{2}$$

Here, N_c is the total number of multipath components, α_i the weight and τ_i the propagation delay of the ith path. $n(t)$ is a Gaussian white noise term accounting for both noise effects in the channel as well as antenna and component noise in

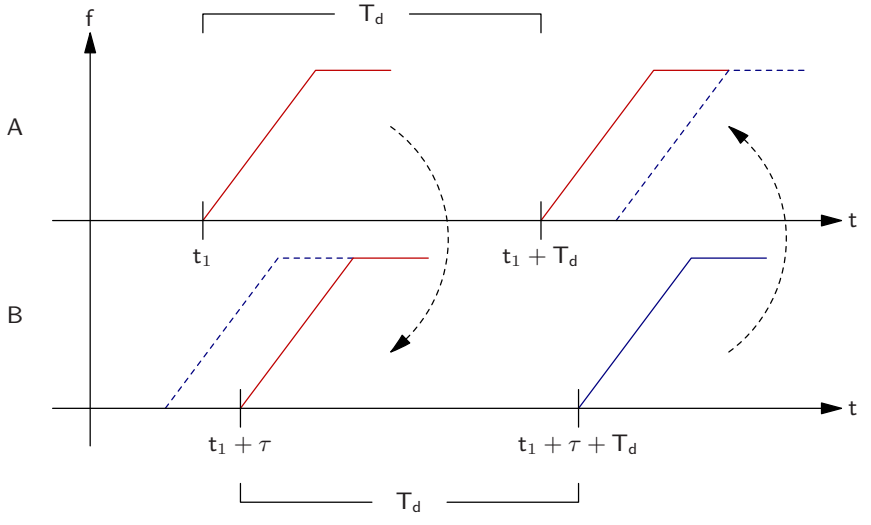


Fig. 1. Outline of the positioning protocol. Station A transmits a ramp at time t_1, which is mixed with an internal, roughly presynchronized ramp in B. After a fixed, known time interval T_d, B retransmits a ramp, which is mixed in A to determine the signal runtime τ.

the transmitter. Note that each path is also subject to a statistical change in excess phase, which can, however, be neglected in the further analysis and is thus not given in (2).

After band filtering and amplification, the incident signal is mixed with a copy of the transmit signal and low-pass filtered. This results in a sum of baseband cosine signals with frequencies corresponding to the path component delay:

$$s_{\mathrm{Mix}}(t) = \sum_{i=1}^{N_c} \tilde{\alpha}_i \cos\big((\mu\tau_i)t + \phi\big) + n_{\mathrm{Mix}}(t) \ . \tag{3}$$

Here, $\tilde{\alpha}_i$ is a modified path amplitude and ϕ a time-invariant phase term. n_{Mix} remains a Gaussian noise term.

After A/D conversion, the baseband signal is windowed and fed into an FFT, which results in

$$S_{\mathrm{RX}}(f) = \sum_{i=1}^{N_c} \bar{\alpha}_i \,\mathrm{sinc}\big(\pi T_o(f - \mu\tau_i)\big) + N(f) \tag{4}$$

for the trivial case of rectangular windowing. Here, $\bar{\alpha}_i$ is the amplitude of the ith component in frequency domain; T_o is the observation length; τ_i the path runtime; and $N(f)$ is a white noise term.

From (4), it is clear that the spacing of components in frequency domain is directly dependent on the ramp slope μ. Choice of the parameters bandwidth

and period is limited by spectral regulations and realtime ability considerations, respectively. The width of the components and the sidelobe attenuation is dependent on the type of window chosen. A Hamming window with a relative attenuation of -42 dB and a frequency domain mainlobe width of $4/T_o$ proves to be a good compromise [4].

Simulations and prototype measurements have shown that the error caused by the overlap of path components is considerable. It was also shown empirically that this error can be significantly reduced by measuring along the slope of the frequency domain shape rather than at the peak. Comprehensive simulations using multipath profiles measured in indoor environments have confirmed that the optimal relative measurement level is strongly dependent on the path profile and noise characteristics of the channel. In the following, we propose a neural network based method to adaptively select the measurement level.

3 Artificial Neural Networks for Level Estimation

Let $\boldsymbol{u} = [u_1 u_2 \dots u_{N_0}]^T$ be a vector of equidistant samples of the noisy channel impulse response (CIR), and y^* be the optimal measurement level, where N_0 denotes the overall number of samples. With the assumption that there exists a nonlinear mapping between the CIR and the optimal measurement level, their relation can be described as

$$y^* = f(\boldsymbol{u}) \quad , \tag{5}$$

where f is the nonlinear mapping function.

In this paper, we propose to employ a multilayer perceptron (MLP), which is a special kind of artificial neural network (ANN), to represent the nonlinear relationship between the sampled CIR and the optimal measurement level. MLPs are pure feedforward neural network structures that are capable of approximating any measurable function to any desired degree of accuracy [5]. Like any other NN, the MLP we employ here consists of neurons (processing elements) and weighted links (connections between the neurons). As shown in Fig. 2, in this special network topology, the neurons are arranged within different layers and connections between the neurons only exist if they are located in adjacent layers. Each neuron can be seen as a summation unit followed by a nonlinear activation function. Under this premise, the output of a neuron can be written as

$$z_i^l = f_{\mathrm{act},i}^l \left(\sum_{j=1}^{N_{l-1}} w_{ij}^l z_j^{l-1} + b_i^l \right) \quad , \tag{6}$$

where z_i^l denotes the output of the ith neuron in the lth layer, z_j^{l-1} with $j = 1 \dots N_{l-1}$ are the output values of the previous network layer and w_{ij} and b_i indicate the connection weights between the jth neuron of the previous layer and the ith neuron of the current layer and an additional bias, respectively. The function $f_{\mathrm{act},i}^l$ denotes the (nonlinear) activation function of the ith neuron. Typical activation functions are sigmoid functions in the hidden layers and linear

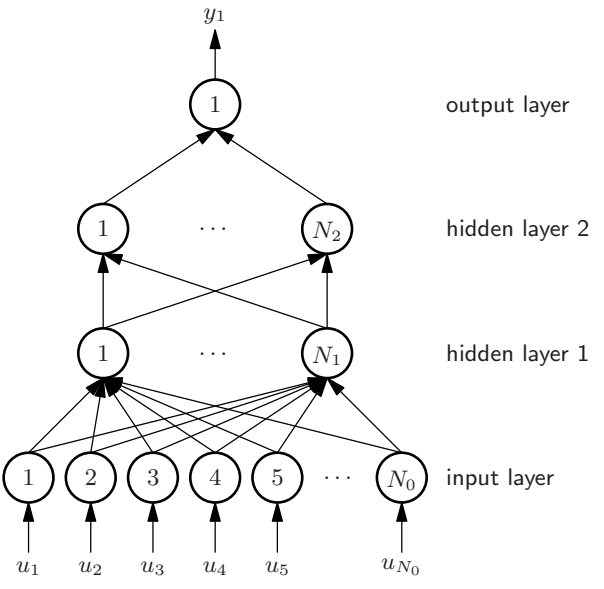


Fig. 2. Drawing of an MLP with two hidden layers as employed in this work. As neuron activation functions nonlinear sigmoid functions for the hidden layers and a linear function for the output layer are used. The neurons of the input layer are special ones and do not have an activation function.

functions in the output layer. The input layer does not consist of neurons with activation functions.

Using an ANN to represent the nonlinear mapping from the CIR on the optimal measurement level, from (5) follows

$$y^* = \tilde{f}\left(\boldsymbol{u}, \mathbf{W}, \boldsymbol{b}\right) \quad , \tag{7}$$

where $\mathbf{W}$ and $\boldsymbol{b}$ denote the weight matrices and the bias vectors, respectively, and $\tilde{f}$ is the overall network function. From Fig. 2, the final formulation of the mapping function follows as

$$y^* = \sum_{k=1}^{N_2} w_{1k}^3 f_{\mathrm{act},k}^2 \left(\sum_{i=1}^{N_1} w_{ki}^2 f_{\mathrm{act},i}^1 \left(\sum_{j=1}^{N_0} w_{ij}^1 u_j + b_i^1 \right) + b_k^2 \right) + b_1^3 \ . \tag{8}$$

An ANN has to be trained before it is able to represent a nonlinear function. This means that through an optimization process, the network weights and biases have to be determined such that the ANN maps any channel profile on the corresponding optimal measurement level. Thus, before the training can be applied, a data set which consists of a few samples $(\boldsymbol{u}_d, y_d^*)|_T$ of channel impulse responses and the corresponding optimal measurement level has to be generated. With this

training data set, the goal of the network training process can be formulated as follows:

$$\mathbf{W}^*, \boldsymbol{b}^* = \arg\min_{\mathbf{W},\boldsymbol{b}} \sum_s \left\| y^*_{d,s} - \tilde{f}\left(\boldsymbol{u}_{d,s}, \mathbf{W}, \boldsymbol{b}\right) \right\| \quad , \tag{9}$$

with $\mathbf{W}^*$ and $\boldsymbol{b}^*$ denoting the optimal weight matrices and bias vectors, respectively, and $\|\cdot\|$ indicating a suitable norm, e.g. the Euclidean norm. The summation index s runs from 1 up to the number of samples contained in the training data set.

In the current work the MLP was trained using the gradient-based backpropagation of error algorithm [6]. Due to the use of this local optimization procedure the initial starting point has to be chosen sufficiently close to the global optimal solution $\mathbf{W}^*$ and $\boldsymbol{b}^*$ to overcome the problem of stepping into local minima. Therefore, a linear-least-square initialization algorithm [7] has been applied that propagates back the desired network response through the layers of the MLP and thus enables an initialization of the network weights and biases in the minimum mean-square-error sense.

After the determination of the starting point for the optimization algorithm the network training could be accomplished. After the optimal weighting values have been found, the functionality of the trained ANN was validated using a few additional test samples $(\boldsymbol{u}_d, y^*_d)|_V$ which were not used during the training process.

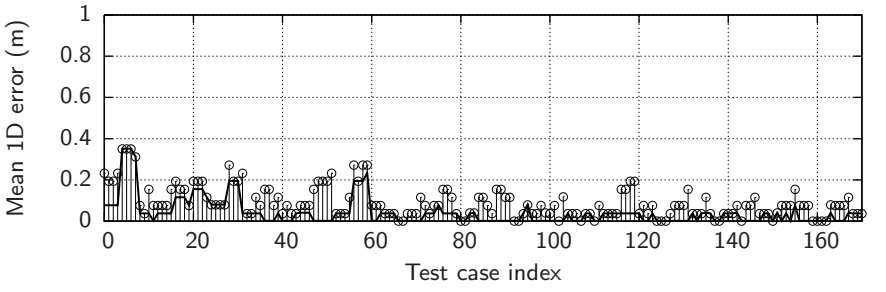
4 Results

Simulations were conducted with a large variety of channel profiles obtained via actual measurements in indoor environments. When we refer to test cases in the following, we always mean a specific combination of channel profile and signal to noise (SNR) ratio setting. For the simulations, all other elements in the signal processing chain were assumed to be ideal, i.e. not causing a mean accuracy error greater than 1 cm in total.

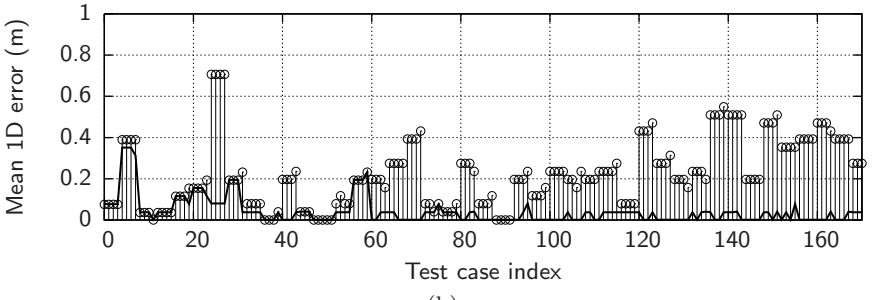
The neural network was trained with a set of constructed CIRs viewed as being representative of a wide variety of indoor scenarios. These CIRs often consisted of as little as two data points different from zero, the line of sight path and a second, weaker component.

To judge the performance of the algorithm, the output of the neural network arbitration was compared to fixed measurement levels oblivious of the specific scenario. A level of 1.0 means brute-force peak detection. Other levels are given relative to the highest amplitude in the normalized frequency domain profile. Figure 3 shows plots of the 1D positioning error achieved by various detection strategies versus the smallest possible error. Note that this error is nonzero due to the fact that there is always a certain amount of overlap with the second path.

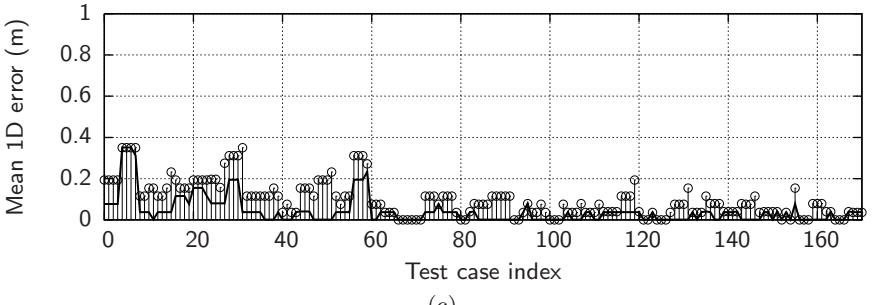
It is obvious that the neural network by far outperforms peak detection. The gain compared to the relative measurement level of 0.4, which was determined by simulations to exhibit the best performance among fixed levels, is less but still significant for a high accuracy system.



(a)



(b)



(c)

Fig. 3. Simulation results for various measurement strategies, with the optimal level (solid line) as reference. (a) Adaptive approach with neural networks (b) Peak detection (c) Relative measurement level of 0.4.

Table 1 compares the performance of the detection strategies by means of three parameters: the mean error μ, the error variance σ^2 and the cumulative deviation from the optimum Σ_D. The latter parameter is simply given by

$$\Sigma_D = \sum_{i=1}^{N_T} \epsilon_{\text{estimated}}(i) - \epsilon_{\text{optimal}}(i) \ , \tag{10}$$

Table 1. Numerical comparison of detection strategy performance

Arbiter	μ	σ^2	Σ_D
ANN	0.0481	0.0028	8.2905
Fixed (0.2)	0.0764	0.0037	13.1582
Fixed (0.4)	0.0571	0.0031	9.8197
Fixed (0.6)	0.0623	0.0031	10.7072
Fixed (0.8)	0.0849	0.0063	14.6230
Fixed (1.0)	0.1849	0.0297	31.7945

where $\epsilon_{\text{estimated}}$ and $\epsilon_{\text{optimal}}$ are the errors caused by the actual arbiter and the minimum error, respectively, and N_T is the number of test cases.

5 Summary

In this work, we have demonstrated the application of artificial neural networks to the problem of path detection in a strong multipath environment. Line of sight path detection is essential for high precision indoor location systems. The error induced by limited bandwidth and thus spectral resolution in an FMCW positioning system can partly be mitigated by adaptively selecting a different measurement threshold. A neural network has been trained to recognize the optimal level in a wide variety of test cases. Simulation results prove our method to be superior to conventional evaluation techniques, where a fixed measurement level is used, at only marginally increased implementation effort.

References

1. Vossiek, M., et al.: Wireless local positioning. IEEE Microwave Magazine 4(4), 77–86 (2003)
2. Ellinger, F., et al.: EU project RESOLUTION. In: International Conference on Wireless Information Networks and Systems (WINSYS), August 2006, pp. 362–366 (2006)
3. Wiebking, L.: Entwicklung eines zentimetergenauen mehrdimensionalen Nahbereichs-Navigationssystems. PhD thesis, Technische Universität Clausthal (December 2002)
4. Oppenheim, A.V., Schaffer, R.: Digital Signal Processing. Prentice-Hall, Englewood Cliffs (1975)
5. Hornik, K.: Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366 (1989)
6. Zhang, Q.J., Gupta, K.C.: Neural Networks for RF and Microwave Design. Artech House, Inc., Norwood, MA, USA (2000)
7. Erdogmus, D., et al.: Linear-Least-Squares Initialization of Multilayer Perceptrons Through Backpropagation of the Desired Response. IEEE Transactions on Neural Networks 16(2), 325–337 (2005)

Object Oriented Signal Data Structures in VLSI Implementations of Wireless Modems

Werner Hein[1], Jens Berkmann[1], Manfred Zimmermann[1], and Mario Huemer[2]

[1] Infineon Technologies, Wireless Communications,
Am Campeon 1-12, D-85579 Neubiberg, Germany
Werner.Hein@infineon.com, Jens.Berkmann@infineon.com,
Manfred.Zimmermann@infineon.com
[2] Klagenfurt University, Embedded Systems Group,
Lakeside B02B, A-9020 Klagenfurt, Austria
mario.huemer@ieee.org

Abstract. In real-world very large scale integration (VLSI) silicon implementations of 3GPP wireless modems modularity of system on chip (SoC) architecture and efficiency in terms of power and die size are crucial criteria. This is particularly true for the signal processing subsystem as the data flow volume in the modem's physical layer is proportionally high.

One architectural method to deal with modularity and power efficiency is to focus on well defined communication structures inbetween the dedicated signal processing units and well formatted data objects stored locally in intermediate buffers to keep data transport effort low. Contrary to that, die size reduction often implies the usage of centralized and shared entities like memories and busses.

This paper presents an example for data communication structure and data format of the radio frame buffers in a UMTS/HSDPA/MBMS wireless modem receive path and how an object oriented system design approach combines those optimization strategies according to the initially mentioned criteria.

1 Introduction

3rd generation mobile phone market - comming from a low volume high end segment - now enters the midrange segment and will sooner or later arrive in mass market. This trend to higher volumes at decreasing prices per unit puts more and more attention onto the design's optimization.

On the other hand in this market still a lot of new services and features are approaching generally to be handled as add-on capabilities which imply modularity and scalability.

1.1 SoC Optimization Requirements ...

From the system architecture's viewpoint SoC optimization in a cost sensitive mobile product can be simplified into two aspects:

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 936–943, 2007.
© Springer-Verlag Berlin Heidelberg 2007

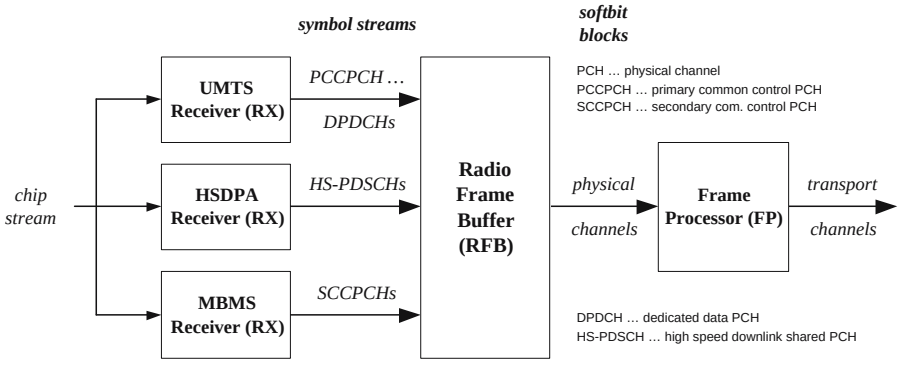


Fig. 1. Typical block diagram for the receive path of a physical layer implementation of a 3rd generation UE. The radio frame buffer is embedded inbetween the various receivers (RX) sending symbol streams and the frame processor (FP) consuming the physical channels framewise and channel-wise.

- Silicon die size and component number minimization - most important driver of user equipment's (UE) production costs.
- Power consumption reduction - relevant driving factor of service availability at UE side.

1.2 ... and Measures

One measure for die size reduction is the minimization of internal static RAM (SRAM) memory exactly there where the data volume to be processed is inevitable high. Additionally, power consumption reduction in signal processing subsystems goes hand in hand with minimization of data transport inbetween the subsystem's units.

Both aspects directly lead to the signal processing subsytem of the UE's physical layer implementation (PHY) as there data volume and data throughput is proportionally high compared against upper layer parts.

The above mentioned term *data* means the payload, which is generally increased by several factors reaching from wordlength for fixpoint signal value representation for numerical robustnes to redundancy added for forward error correction by channel coding.

2 Object Oriented Approach for the Radio Frame Buffer

2.1 Radio Frame Buffer Within Receiver Path

As given in Fig. 1 the radio frame buffer (RFB) receives concurrently the output symbol streams of the various physical channels. In the RFB the physical channels (PCH) get segmented frame-wise and stored until the frame processor (FP)

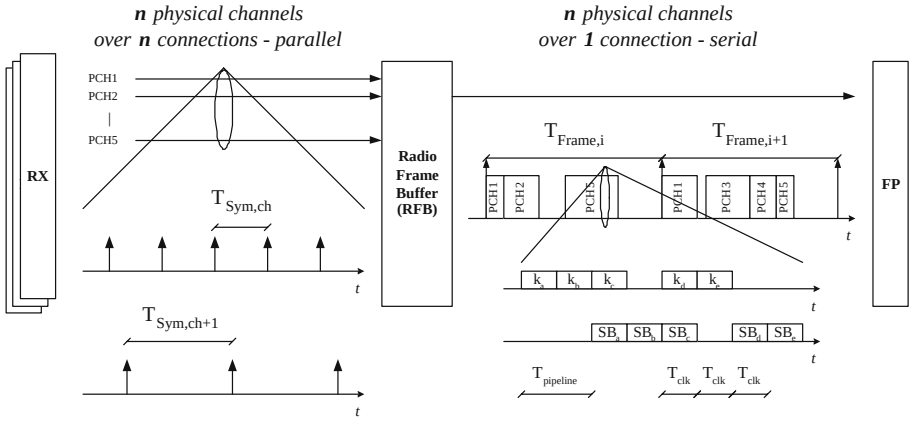


Fig. 2. Illustration of the common properties of input written to and output read from RFB. Input transactions of parallel physical channels occur periodically at stream-typical inter-symbol-intervals. Output transactions are requested sequentially channel block by channel block - and randomly within channel block. Request and delivery are pipelined at a depth of T_{pipeline}.

consumes the PCHs block-wise. The data input per single PCH is periodic at inter-symbol-rate $R_{\text{Sym},ch,i}$ and fairly low compared to the throughput capacitiy of a clocked SRAM. This means that

$$1/R_{\text{Sym},ch,i} = T_{\text{Sym},ch,i} >> T_{\text{clk}} \quad \forall ch, i \ . \tag{1}$$

But the RFB has to receive the PCHs in parallel and the PCHs' symbol rates are different and even each PCH's rate may vary from frame to frame dependent on the PHY's configuration. Additionally, different PCHs may have different frame periods $T_{\text{Frame},i}$ and even PCHs with the same frame period may be shifted to each other. Therefore we have to ensure by design that for any time interval the memory's access rate

$$R_{\text{MEM}} = 1/T_{\text{clk}} > \sum_{ch} R_{\text{Sym},ch,i} \ . \tag{2}$$

Whenever a frame of a single PCH is completed within the RFB this block is scheduled for further processing by the FP. The FP itself is assumed to processes the PCHs sequentially in the order the frame blocks arrive according to *First-come-first-served scheduling policy* from Sect. 1.2.3 in [1].

Contrary to the receivers' (RX) access behaviour the FP reads the softbits (SB) from a PCH block in a random-like manner and as fast as the internal processing performance consumes the SBs (typically $1SB/T_{\text{clk}}$).

Figure 2 shows in general the differences between the data input to and the data output from the RFB.

2.2 Object Oriented Approach

An SoC architecture design methodology following the principle of modularity and scalability shall make use of two essential properties of object orientation:

- ***Data encapsulation*** keeps data storage and handling (RFB) separate from data producers (RX) and data consumer (FP) - this enables to find memory utilization strategies exploiting PCH properties like data persistance effects for worst case memory size determination as well as memory control algorithms for dynamic and power efficient segment management.
- ***Polymorphism*** disengages data consumer (FP) from the case sensitive task of transforming producers' (RX) data into a consumable format. This includes e.g. symbol to softbit translation, provides a PCH property independent data access method for the FP and minimizes the number of data move transactions.

Applying these two properties to the RFB problem not only keeps SoC design modular but concurrently helps to solve above mentioned optimization problems.

3 Design for Data Encapsulation

In this section the data objects of the RFB as well as the corresponding interfaces are defined.

3.1 Object Definitions

Input Object: Common to all RX is that for each PCH ch every i^{th} frame $T_{\mathrm{Frame},ch,i}$ a sequence of $N_{ch,i}$ complex valued symbols

$$\mathbf{S}_{ch,i} = \begin{pmatrix} \underline{S}_{ch,i,0} \\ \vdots \\ \underline{S}_{ch,i,N_{ch,i}-1} \end{pmatrix} \tag{3}$$

is written into the RFB with a period of $T_{\mathrm{Sym},ch,i}$ between symbols. The $\mathbf{S}_{ch,i}$ vector's properties just depend on channel number ch and frame number i.

From an implementation viewpoint an RX just has to signal these two parameters before it pipes the symbols in an equidistant manner into the RFB like into a FIFO. Therefore a typical input object for a $\mathrm{PCH}_{ch,i}$ can be defined by

$$\mathbf{I}'_{ch,i} = [ch, i, \mathbf{S}_{ch,i}] \quad . \tag{4}$$

Output Object: Once such an input object $\mathbf{I}'_{ch,i}$ is completely stored in the RFB and additionally the RX has detected the PCH's transport format this object is scheduled for further processing by the FP.

There are two major changes how the FP sees the stored PCH frame object:

1. The FP wants to read softbits,
2. and wants to read them in a de-interleaved order.

De-interleaving means that the previous vector's $\mathbf{S}_{ch,i-1}$ elements have to be re-ordered by

$$\mathbf{S}'_{ch,i} = \mathbf{D}_{ch,i} \bullet \mathbf{S}_{ch,i-1} \tag{5}$$

where $\mathbf{D}_{ch,i}$ is generally a permutation matrix of dimension $N_{ch,i} \times N_{ch,i}$ which can be derived for each frame according to sections 4.2.11 and 4.5.6 of [2] and which elements d_{ij} of $\mathbf{D}_{ch,i}$ are either 0 or 1.

Depending on the PCH's modulation scheme $m_{ch,i}$ defined in section 5.1.1 of [3] the softbit vector $\mathbf{SB}_{ch,i}$ of size L can be derived which elements $SB_{ch,i,k}$ can be extracted from $\mathbf{S}'_{ch,i}$ in (5) by

$$SB_{ch,i,k} = \begin{cases} \text{if } m_{ch,i} = \text{QPSK}, K = \{k \in N_0^+ \ k < L = 2 \cdot N_{ch,i}\} \\ \quad \begin{cases} \Re\left\{\underline{S}'_{ch,i,k/2}\right\} & \forall k \in K \wedge (k \bmod 2) = 0 \\ \Im\left\{\underline{S}'_{ch,i,(k-1)/2}\right\} & \forall k \in K \wedge (k \bmod 2) = 1 \end{cases} \\ \\ \text{if } m_{ch,i} = \text{16QAM}, K = \{k \in N_0^+ \ k < L = 4 \cdot N_{ch,i}\} \\ \quad \begin{cases} q \cdot \Re\left\{\underline{S}'_{ch,i,k/4}\right\} & \forall k \in K \wedge (k \bmod 4) = 0 \\ q \cdot \Im\left\{\underline{S}'_{ch,i,(k-1)/4}\right\} & \forall k \in K \wedge (k \bmod 4) = 1 \\ p - q \cdot |\Re\left\{\underline{S}'_{ch,i,(k-2)/4}\right\}| & \forall k \in K \wedge (k \bmod 4) = 2 \\ p - q \cdot |\Im\left\{\underline{S}'_{ch,i,(k-3)/4}\right\}| & \forall k \in K \wedge (k \bmod 4) = 3 \end{cases} \end{cases} \tag{6}$$

where p and q can be derived from Table 3B in [3]. Now we can define the output object for a $\text{PCH}_{ch,i}$

$$\mathbf{O}_{ch,i} = [ch, i, \mathbf{SB}_{ch,i}] \ . \tag{7}$$

From the implementation's point of view the de-interleaving matrix $\mathbf{D}_{ch,i}$ is intentionally applied to the RFB when the FP applies the index k for softbit $SB_{ch,i,k}$ and need not be included into any object considered here. But of course the modulation scheme $m_{ch,i}$ has to be integrated and the input object has to be extended to

$$\mathbf{I}''_{ch,i} = [ch, i, m_{ch,i}, \mathbf{S}_{ch,i}] \ . \tag{8}$$

3.2 Internal Data Organization (An Example)

After definition of the objects nature delivered to the input ports and retrieved from the output port Fig. 3 gives an example how the data could be organized within the RFB's memory. This organization shall support

1. reduction of the memory footprint - in this example done by circular re-use of the buffer in which a specific PCH_{ch} is written in, and
2. easy to handle transformations between the several representations of $\text{PCH}_{ch,i}$ which will be addressed in Sect. 4.

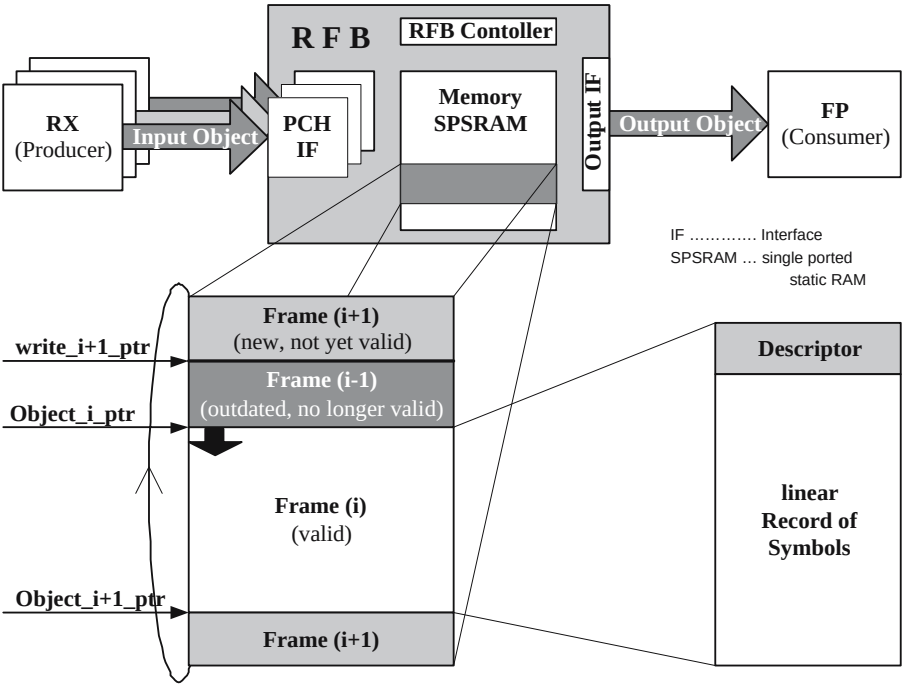


Fig. 3. Cyclic Data Organisation: newer objects overwrite outdated - already consumed - objects. Information for data management and data format description are placed as header in front of the linearly written symbol data.

4 Design for Polymorphism

In this section the different representations for one and the same object $\mathrm{PCH}_{ch,i}$ are addressed and it is described how they are translated within real time for the multiple input ports and with maximum throughput for the output port. Translation can be split into two parts - the transformation of the signal data values and the signal data localization, respectively. Figure 4 shows as an example a typical implementation of the internal data flow.

4.1 Translations

Value Translation: Unfortunately in real life the received signal is distorted by noise which results in a fuzzy distribution of the symbols in the signal map (see section 6.5.2 in [4]).

To correct this the average deviation is measured during reception typically each timeslot s and included into the symbol-to-softbit mapping in (6).

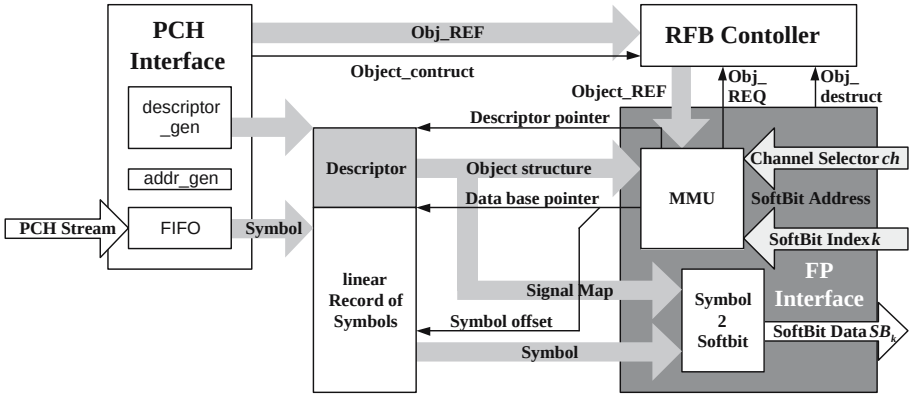


Fig. 4. Data Translations: on the way of an object from RXs as producers via RFB towards FP as a consumer various translations take place.

$$\mathbf{C}_{ch,i} = \begin{pmatrix} p_{ch,i,0} & q_{ch,i,0} \\ \vdots & \\ p_{ch,i,M-1} & q_{ch,i,M-1} \end{pmatrix} \tag{9}$$

where M is the number of slots per $\mathrm{PCH}_{ch,i}$ frame depending on the frame interval $T_{\mathrm{Frame},ch,i}$

$$M = \begin{cases} 3 & \text{if } T_{\mathrm{Frame},ch,i} = 2\mathrm{ms} \\ 15 & \text{if } T_{\mathrm{Frame},ch,i} = 10\mathrm{ms} \end{cases} . \tag{10}$$

With (9) and (10) we have to refine (8) to

$$\mathbf{I}'''_{ch,i} = [ch, i, m_{ch,i}, \mathbf{S}_{ch,i}, \mathbf{C}_{ch,i}] \quad . \tag{11}$$

Finally this translation has to be done in the block *Symbol-2-Softbit* as depicted in Fig. 4 within a given time budget T_{pipeline} whenever the FP reads a specific SB_k.

Reference Translation: Already when an RFB implementation uses a quite basic memory size reduction technique like given in Sect. 3.2 a memory management functionality has to be foreseen to translate the co-ordinates ch and k of $SB_{ch,i,k}$ applied by the FP into all the physical memory locations for input to the value translation described above. Furthermore this translation done in the block *MMU* in Fig. 4 has to fit into the same time budget T_{pipeline} given in (2).

4.2 Multiport Access and Residual Bandwidth

In today's VLSI chip design implementations of real dual ported SRAM memories (DPSRAM) which allow concurrent write and read access to the same

locality are avoided and whenever possible replaced by a single ported SRAM (SPSRAM) because of technological advantages.

However, in our case even a dual ported memory is not sufficient as there are typically two separated RX blocks and one FP in today's modem designs. One RX is usually a RAKE receiver sending asynchronous PCH symbol streams for PCCPCH, SCCPCH and DPDCH. For design modularity reasons this receiver can be split up into more than one block to e.g. deal with different capability classes (e.g. separate MBMS support as in Fig. 1). The other receiver very often is of equalizer-based type and demodulates the HS-PDSCHs which are always frame- and symbol-wise synchronous.

Because of those multiple ports and the already in Fig. 2 mentioned different behaviours at the interfaces we need an access arbiter in front of the SPSRAM.

From (8) it is obvious that allmost all parts of the input objects are piped into the RFB in a regular manner. Only the output port needs to maintain a clock cycle based behaviour as the FP requires to access the data with maximum throughput. Meeting the real time requirements for all inputs and due to the SPSRAM approach for the RFB memory the average access bandwidth of the FP degrades according to (1) and (2) to

$$R_{\mathrm{FP},i}^{\mathrm{avg.}} = R_{\mathrm{MEM}} \cdot (1 - T_{\mathrm{clk}} \cdot \sum_{ch} 1/T_{\mathrm{Sym},ch,i}) \ . \tag{12}$$

5 Conclusion

Even for pure digital signal processing applications implemented on silicon a data centric approach including data organisation and well-founded access methods supports SoC design in its two most demanding topics - modularity and optimisation. But of course object orientation is not for free and as described in this paper extra effort has to be taken for a proper object design to gain the revenue.

References

1. F. Cottet, J. Delacroix, C. Kaiser, Z. Mammeri; Scheduling in Real-Time Systems; Wiley (2002)
2. 3GPP:TS 25.212 - 3rd Generation Partnership Project; Technical Specification Group Radio Access Network;
   Multiplexing and channel coding (FDD), Release 6 (2006-12)
3. 3GPP:TS 25.213 - 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Spreading and modulation (FDD), Release 6 (2006-03)
4. Edward A. Lee, David G. Messerschmitt; Digital Communication; Kluwer (1988)

A Bandwidth Efficiency Optimized Frequency Domain Equalization Concept for Single Carrier Transmission

H. Witschnig[1], H. Stallinger[2], and M. Huemer[3]

[1] NXP Semiconductors, Gratkorn, Austria
harald.witschnig@nxp.com
[2] Institute for Communications and Information Engineering, University of Linz, Austria
[3] Embedded Systems Group, Klagenfurt University
mario.huemer@uni-klu.ac.at

Abstract. The concept of Single Carrier Transmission with Frequency Domain Equalization (SC/FDE) represents one of the most powerful strategies for next generation wireless communication systems. One of the main advantages of this concept, which makes use of a so called cyclic prefix (CP), is its powerful and low complexity equalization procedure in frequency domain. In this paper we investigate frequency domain equalization schemes based on the more bandwidth efficient concepts of Overlap and Add (OA) and Overlap and Save (OS). We explore the main problem of these approaches, namely the difficult predefinition of a maximum equalizer length, and we present a solution which results in a bit error performance absolutely comparable to the CP based equalizer.

1 Introduction

The concept of frequency domain equalization represents one of the most powerful strategies to combat time dispersion caused by multipath propagation. Two well known concepts exist: OFDM (Orthogonal Frequency Division Multiplexing) and SC/FDE. While the interest on OFDM was almost overwhelming the last ten years, the interest on SC/FDE has been growing rapidly only recently. SC/FDE features advantages especially for the uplink transmission in mobile equipment. Fig. 1 gives an overview of the concepts of frequency domain equalization focusing on single carrier schemes, which are topic of the underlying investigations. The concepts of SC/FDE are distinguished between the strategies of OA and OS [1] on the one hand side and CP on the other hand [2]. Up to now the concepts of OA/OS were almost completely neglected for high data rate wireless communications. But we will demonstrate in this work that these strategies are comparably powerful in terms of bit error performance, but feature higher bandwidth efficiencies compared to the CP approach. Besides this classification, the CP-scheme can additionally be separated between a classical structure and a structure using a so called Unique Word (UW) based on known pilot-symbols [3],[4]. This UW can advantageously be used for additional purposes like equalization, synchronization etc..

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 944 – 951, 2007.
© Springer-Verlag Berlin Heidelberg 2007

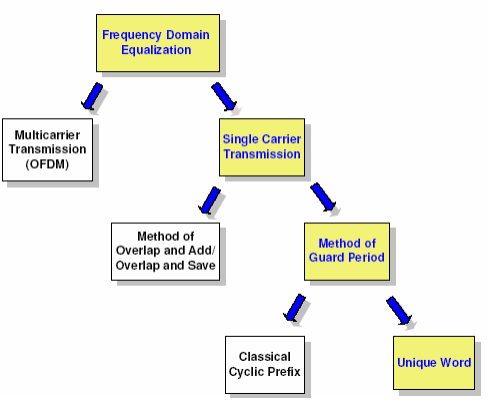


Fig. 1. Classification of frequency domain equalization concepts

2 SC/FDE Concepts

The theorem of (linear) convolution is of essential significance for all further investigation. It states, that the time domain convolution of two functions *x(t)* and *h(t)* is transformed into the algebraic product of the two corresponding Fourier transformed *X(f)* and *H(f)*:

$$x(t) * h(t) \quad \circ\!-\!\bullet \quad X(f) \cdot H(f) \tag{1}$$

The theorem of (linear) convolution allows to replace the costly convolution operation by a multiplication operation in the frequency domain. In practical applications Fourier transforms of finite-length sequences are carried out by the help of the DFT (Discrete Fourier Transformation). Comparable to the theorem of linear convolution for the continuous Fourier transformation the theorem of cyclic convolution for the DFT can be formulated as [1]

$$x[n] \otimes h[n] \quad \circ\!-\!\bullet \quad X[k] \cdot H[k] \tag{2}$$

Here $x[n]$ and $h[n]$ are discrete-time sequences, and $X[k]$ and $H[k]$ represent the corresponding discrete Fourier transformed sequences. We conclude that the cyclic convolution of two sequences is equivalently represented by the algebraic product of the discrete Fourier transformed sequences as

$$x[n] \otimes h[n] = \mathrm{IDFT}_N \{\mathrm{DFT}_N \{x[n]\} \cdot \mathrm{DFT}_N \{h[n]\}\} \tag{3}$$

Here N is the maximum length of $x[n]$ and $h[n]$. Although equation (3) represents the most essential relation in the context of frequency domain equalization, its direct use is not possible in that context as the distortion by the radio channel and the equalization by an FIR (Finite Impulse Response) filter are represented by a linear convolution but not by a circular convolution.

In the following concepts will be demonstrated that allow the use of the efficient DFT and to perform a correct linear filtering in the frequency domain in spite of the problem mentioned above. First we discuss a frequency domain methodology that is

equivalent to linear time domain convolution, in the following we discuss the CP-approach, which is based on the circular convolution.

2.1 The Concept of Overlap and Save (OS)

The straight forward approach of a linear FIR-equalizer implementation in frequency domain is to transform the received sequence to frequency domain and to perform a multiplication operation with the equalizer frequency response for each frequency tap. For that we would need large DFTs, and processing could not start until the whole sequence is received. Therefore the received sequence is reasonably processed blockwise. The two well known concepts of OA and OS solve this problem efficiently, interestingly they have only rarely been investigated for frequency domain equalization approaches. Exemplary the concept of OS is described in detail, while for OA we refer to [1].

For Overlap-Save the size of a processed block is $N=L+M-1$. Processed means that an N-point DFT is carried out, followed by multiplication operations and an N-point IDFT. Here each data-block consists of L new data values of the actual block and of $M-1$ old data values of the previous block to guarantee the correct processing of the influence of symbols of the previous block (inter-block-interference). M corresponds to the maximum time domain equalizer length. Fig. 2 demonstrates the data structuring and the processing approach. Note that for the first processed block $M-1$ zeros are prefixed.

The Fourier transformed sequences of the blocks $r_i[n]$ are given by

$$R_i[k] = \sum_{n=0}^{N-1} r_i[n] e^{-j2\pi kn/N}, \qquad k = 0,...,N-1 \tag{4}$$

In general the equalizer frequency response $E[k]$, $k=0,\ldots,N-1$ is obtained by a DFT operation of the zero padded sequence of the (originally M) time domain filter-taps. For each block the equalization is carried out by

$$\hat{Y}_i[k] = E[k] \cdot R_i[k], \qquad k = 0,...,N-1 \tag{5}$$

An N-point IDFT leads finally to the equalized block

$$\hat{y}_i[n] = 1/N \sum_{k=0}^{N-1} \hat{Y}_i[k] e^{j2\pi kn/N}, \qquad n = 0,...,N-1 \tag{6}$$

As the first $M-1$ samples of $\hat{y}_i[n]$ are corrupted by aliasing (due to the cyclic convolution property of the DFT), they have to be discarded (note these $M-1$ samples were already equalized correctly in the previous block). The last L samples of the actual processed block represent the i^{th} equalized data block $y_i[n]$:

$$y_i[n] = \hat{y}_i[n+M], \qquad n = 0,...,L-1 \tag{7}$$

Putting all blocks together, the whole process corresponds to a linear convolution of the received and distorted data stream with the equalizer impulse response.

For this approach it is essential to define the overlap M appropriately. Obviously M should correspond to the time domain equalizer length. The problem is that the

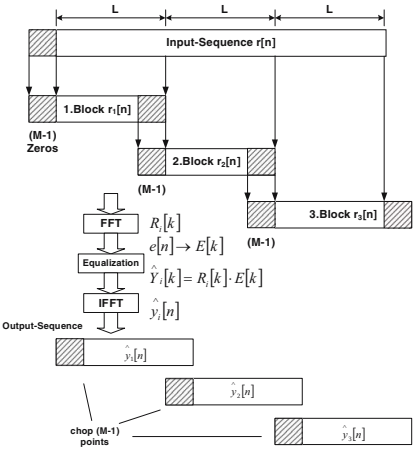


Fig. 2. Concept of Overlap and Save

equalizer length has to be known in advance, which is typically not the case in wireless transmission. Furthermore, due to the time varying behavior of the wireless channel the equalizer, and consequently also the equalizer length, is not constant from burst to burst. Theoretically the equalizer length may vary from one (in the case of an Additive White Gaussian Noise (AWGN) channel) to infinity, depending on the actual channel to be equalized. Predefined equalizer lengths, which lead to satisfying results for all channel scenarios, do not exist. In [5] it has been recommended to choose *M=N/2.* While this definition seems to generate much processing overhead for some AWGN-like channel scenarios, it might constitute a (too) strong restriction for other scenarios. Nevertheless, the actual investigations are based on this assumption.

2.2 The Cyclic Prefix Based Concept

The strategy of OA/OS is to solve a linear convolution problem by a blockwise application of cyclic convolutions (realized in frequency domain) together with different zero padding and overlap strategies. A completely different approach to equalize a channel distorted data stream with the use of DFTs is the CP-scheme [2]. The idea behind is not only to apply blockwise processing at the receiver side as for OA/OS but also to perform a blockwise transmission by inserting a guard interval (cyclic prefix) between successive blocks. In order to mitigate interblock interference the duration T_G of the guard interval has to be longer than the duration T_h of the channel impulse response $h(t)$. Fig. 3 depicts the transmit data structure. Each block consists of a data part of duration T_{FFT} (composed by L data symbols) and the cyclic extension with duration T_G. Due to the cyclic extension of the transmitted blocks, the convolution of one cyclically extended transmitted block and the channel impulse response $h(t)$ corresponds to the circular convolution of the original data block $s_i(t)$ (the original, not cyclically extended block) and $h(t)$. This corresponds to the frequency domain relation

$$R_i[kf_0] = H[kf_0] \cdot S_i[kf_0] \tag{8}$$

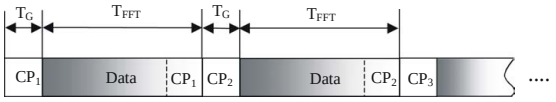


Fig. 3. Data structure for the cyclic prefix based transmission scheme

for $k \in \mathbf{Z}$ and $f_0=1/T_{FFT}$. Here the functions $R_i(f)$, $S_i(f)$ and $H(f)$ are related to the time domain signals $r_i(t)$ (one period of the received data block), $s_i(t)$ (the original, not cyclically extended block) and $h(t)$. Equation (8) represents the cyclic convolution theorem for continuous-time signals.

Compared to the OA or OS concepts, the following advantages can be identified:

• A zero padding or overlap as for OA/OS is not necessary, as the information is ”concentrated” within one processed block. This allows to use a DFT of reduced order, resulting in a lower signal processing complexity.

• A pre-assumption of the equalizer length, as it is necessary for OA/OS, is completely prevented. In contrast to the overlap in OA/OS the guard duration is not depending on the equalizer length but on the channel duration, which is much more predictable for a specific application scenario. Furthermore, the CP structure allows to implement an optimal equalizer of infinite length.

The price to be paid for these significant advantages is a reduced bandwidth efficiency of up to 20% as the guard period has to be inserted already at the transmitter.

3 Performance Comparison: CP versus OS

Let us first compare the OA/OS schemes with the CP approach for an AWGN transmission. For that case the CP schemes suffer from a loss, which results from the fact that the effective energy per data-symbol is reduced by the symbols of the guard period (which do not carry information). Based on a white Gaussian noise with the two sided power spectral density of $N_0/2$ the S/N ratio results in

$$\frac{S}{N} = \frac{E_b}{N_0/2}\left(1 - \frac{T_G}{T_{FFT}}\right) \tag{9}$$

Depending on the ratio of T_G and T_{FFT} the typical performance loss is in the range between 0.6 and 1 dB.

On the other hand, as pointed out in the last section, the concept of OA/OS will suffer from a performance loss compared to the CP scheme, as soon as the equalizer length exceeds the pre-defined overlap length M. It is aim of this investigation to characterize the expected performance degradation and in particular to characterize under which circumstances this degradation arises. For that we studied the behavior of the different schemes for a variety of channel snapshots. Exemplary we show the detailed results for two particular channel characteristics (that we will call channel A and channel B). Note that we assumed an ideal channel estimation (performed in frequency domain), from that we are able to easily calculate the ideal equalizer frequency response. The approach to obtain the appropriate windowed equalizer frequency response for the OA/OS schemes was as follows:

- Calculation of the equalizer impulse response by applying an IDFT to the ideal equalizer frequency response.
- Windowing of the equalizer impulse response with a rectangular time domain window of length *M*.
- Transformation back to frequency domain to obtain the equalizer frequency response finally used for the OA/OS approach.

Exemplary Fig. 4 shows the results for channel A which features a comparably low frequency selectivity (no deep spectral fades appear in the channel frequency response). From Fig. 4b we recognize, that the equalizer impulse response fades away within the predefined window length *M*. Consequently there is no obvious difference between the original ideal equalizer frequency response (Fig. 4a) and the frequency response obtained via the procedure described above (Fig. 4c). Fig. 4d depicts the BER (bit error rate) for OA/OS as well as for CP. The main conclusion is, that OA/OS performs about *1 dB* better than CP, as it is also the case for AWGN transmission. The restriction of the equalizer length for OA/OS has no negative effect on the performance. This in general holds for channels with low frequency selectivity. Another well known fact can be seen in Fig. 4d, namely that the performance gain of Minimum Mean Square Error Equalization (MMSE) compared to Zero Forcing (ZF) is small for that class of channels.

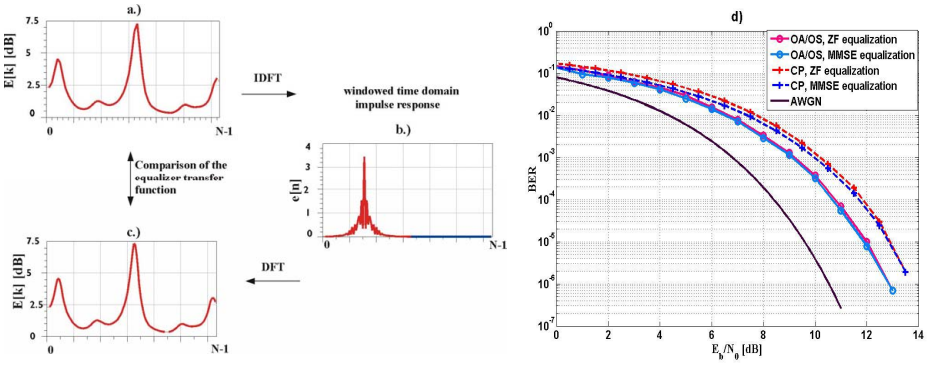


Fig. 4. Generation of the equalizer frequency response for OA/OS, and performance comparison in terms of BER (channel A)

This behavior changes significantly for channel B, which features deep spectral notches. Fig. 5 demonstrates, that now the windowing process for the equalizer impulse response significantly changes the finally resulting equalizer frequency response. The impulse response makes obvious that there is a dominant oscillation corresponding to the deep spectral fade. Fig. 5b) clearly shows, that for channel B the impulse response definitely not fades away within the predefined window length. The bit error behavior for the OA/OS schemes is presented in Fig. 5d). A simple ZF equalizer is completely unable to equalize the distorted information. In comparison to this, the essentially more powerful MMSE equalizer shows a performance improvement as long as the additive term in the denominator of the MMSE equalizer

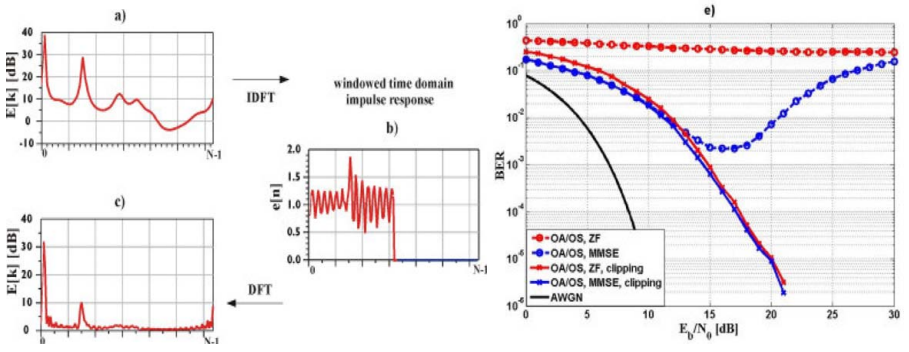


Fig. 5. Generation of the equalizer frequency response for OA/OS, and performance comparison in terms of BER (channel B)

contributes (see [4] for ZF and MMSE equalizer representations). For higher E_b/N_0, the ZF and MMSE criterion pass into each other, resulting in a similar, but unfeasible performance. We learn that a reliable equalization based on OA/OS for channels with deep spectral fades is impossible. The restriction of the equalizer length given by the predefined window length M is too strong for channels with deep spectral fades.

One possible solution for the problem is to increase M significantly. But this approach suffers from two major problems: 1) A general valid predefinition of M for all supposable channel profiles is hardly possible. 2) The signal processing complexity would grow significantly.

In this work a different solution is developed, leading to tremendous improvements and keeping the concept of OA in discussion. It has been demonstrated that the reasons for the performance loss are deep spectral fades in the channel frequency response. A simple but very efficient solution for this problem is to restrict the corresponding high amplifications of the equalizer frequency response (which occur at the same frequencies as the notches appear in the channel response) by a limiter. This procedure is also called clipping. The most essential result of this procedure, proved for a variety of channel snapshots, is that the time domain representation of this amplitude-restricted frequency response typically fades away much faster, which makes it much easier to choose a proper (and rather small) window length M. The effect of this simple limiting process in terms of bit error performance is tremendous as depicted in Fig. 5d). The saturation effect is completely prevented and an acceptable performance results.

Based on the previous investigations, a final performance comparison between the OA/OS schemes and the CP approach has been carried out based on averaging over a total number of 100 indoor radio channel snapshots. Fig. 6 represents the resulting BER performance for MMSE equalization. For the OA/OS schemes the limiter was fixed to a constant level of 20 dB – a value resulting from an optimization process done via Monte Carlo simulations. The optimum value may vary for different channel characteristics, but the chosen value represents a reliable adjustment for all actual channel characteristics. In contrast to OA/OS a limiter is of no meaning for CP, as the MMSE criterion represents an optimum criterion allowing an infinite long equalizer impulse response. Therefore the ideal equalizer frequency response can be applied for

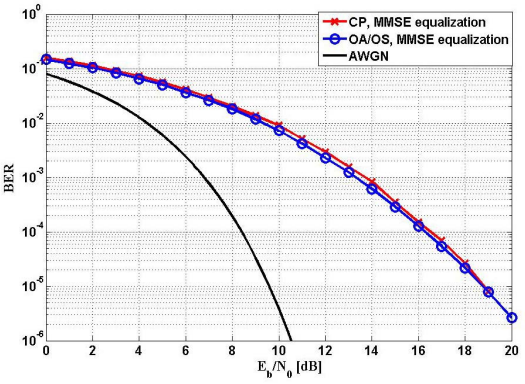


Fig. 6. Performance comparison between OA/OS and CP based on a variety of randomly chosen radio channel snapshots

the CP approach. The obtained results in Fig. 6 lead to the final statement that the performance of CP - and OS based processing is absolutely comparable.

4 Conclusion

Single carrier systems with frequency domain equalization represent a powerful scheme for future high rate communication systems. In this paper we compared the performance of Overlap and Save versus Cyclic Prefix based equalizer concepts. It has been demonstrated that the performance of OS based equalizers depends strongly on the underlying radio channel. This strong dependency has been overcome by the introduction of a so called clipping or limiter function. By applying this idea it was demonstrated that the overall performance of OS and CP based equalizer concepts is now absolutely comparable, while the concept of OS shows the advantage of a higher bandwidth efficiency.

References

1. Proakis, J.C., Manolakis, D.G.: Digital Signal Processing, 3rd edn. Prentice Hall, Upper Saddle River, New Jersey (1996)
2. Sari, H., Karam, G., Jeanclaude, I.: Frequency-Domain Equalization of Mobile Radio and Terrestrial Broadcast Channels. In: Proceedings of the IEEE International Conference on Global Communications (Globecom 1994), San Francisco, USA, pp. 1–5 (1994)
3. Ariyavisitakul, S.L., Eidison, A.B.-S.B., Falconer, B.: Falconer: Frequency Domain Equalization for Single-Carrier Broadband Wireless Systems. White Paper – online available at http://www.sce.carleton.ca/bbw/papers/ Ariyavisitakul.pdf
4. Witschnig, H.: Frequency Domain Equalization for Broadband Wireless Communication - With Special Reference to Single Carrier Transmission Based on Known Pilot Sequences. Ph.D. Thesis, Institute for Communications and Information Engineering, University of Linz, Austria (2004)
5. Sommen, P.C.W., Jayasinghe, J.A.K.S.: On Frequency Domain Adaptive Filters using the Overlap-add Method. In: IEEE Proc. ISCAS, Espoo Finland, pp. 27–30 (June 1988)

Optimized Mapping Schemes for LDPC Coded Higher Order Modulated QAM Transmission

Michael Lunglmayr and Jens Berkmann

Infineon Technologies AG
Am Campeon 1-12, D-85579 Neubiberg, Germany
{michael.lunglmayr,jens.berkmann}@infineon.com

Abstract. Low Density Parity Check (LDPC) codes are considered to be used in many future communication systems. We will show in this work how the performance of a communication system is influenced when different bit to symbol mapping schemes of a higher order modulation (e.g.: 16-QAM) are combined with LDPC coding. To evaluate and optimize bit-to-symbol mapping methods, a modified version of density evolution is proposed. In addition, simulation results are shown which justify the use of density evolution as a method for qualitative analysis of different symbol mapping schemes in conjunction with LDPC coding. We propose novel mapping methods and show performance results in combination with LDPC coding.

Keywords: Channel coding, low-density parity-check (LDPC) codes, density evolution, receiver design, optimization.

1 Introduction

Low Density Parity Check Codes, although invented 1963 by Gallager [1], gained a lot of popularity during the last years. These codes are proposed for many standards of future communication systems e.g. WLAN 802.11n [2]. Many of these systems additionally use higher order modulation such as 16-Quadrature Amplitude Modulation (16-QAM) or 64-QAM to increase the data rate of the system. In these modulation schemes K-tuples of bits are mapped to modulation symbols at the transmitter. At the receiver a demapping process has to be done which demaps the received data sample to log likelihood ratios (LLR's). These LLR's are then used as an input to the decoder of an error correcting code such as an LDPC code. The disadvantage of higher order modulation schemes is that they are much more error prone than for example Binary Phase Shift Keying (BPSK). The error probability of the demapped bits (respectively the LLR values) is greater than the error probability of simpler modulation schemes. In this work the *structure* of the considered codes is used to form bit-to-symbol mapping rules. It is commonly known that, when using LDPC codes, different bit-to-symbol mapping schemes can have an impact on the bit- and block error rates (BER and BLER, respectively) after the decoder. We will show how *density evolution* can be used as a tool to evaluate and optimize these different mapping

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 952–959, 2007.
© Springer-Verlag Berlin Heidelberg 2007

methods. Interestingly, a similar use of density evolution was done to optimize degree distributions of LDPC codes for OFDM in [3].

In this work we will concentrate our focus on quadrature amplitude modulation (QAM). The examples in this article all use 16-QAM. But the concepts shown here can also be applied to other modulation techniques (e.g. phase shift keying; PSK).

2 Higher Order Modulation

In higher order modulation data is grouped in blocks of K bits. A block of K bits is then mapped to one modulation symbols which is transmitted through a communication channel to a receiver. For example in 16-QAM 4 bits are mapped to one modulation symbol. It can be easily shown (e.g. in [4]), that the bit error probabilities of the bits within one symbol are different, when transmitted through an AWGN channel at a given signal-to-noise ratio (SNR). For example for 16-QAM mapping, two bits of the 4-tuple of bits corresponding to a modulation symbol have a smaller bit error probability than the other two bits. So the bits at those two positions can be transmitted more reliably than the bits at the other two positions of a modulation symbol.

The idea of the work described in this article is to use this behavior to improve the overall decoding performance of a system combining modulation and LDPC coding.

3 LDPC Codes

An LDPC code can be described by its parity check matrix $\mathbf{H}$. This matrix describes the valid code words of the code. For every code word $\mathbf{y}$ applies:

$$\mathbf{Hy} = 0. \tag{1}$$

The matrix $\mathbf{H}$ can be used to generate a graph representing an LDPC code, a so called Tanner graph. Fig. 1 shows a parity check matrix of an LDPC code and its corresponding Tanner graph (for simplicity the parity check matrix shown here is much smaller than the parity check matrix of practically used LDPC codes). A tanner graph is a bipartite graph. The nodes of this graph can be divided into two classes. One class is called check nodes (C-nodes, drawn as squares) the other class is called variable or bit nodes (V-Nodes, drawn as circles) as shown in Fig. 1. The C-nodes correspond to the rows of the matrix $\mathbf{H}$ (e.g. C_1 corresponds to row 1 of $\mathbf{H}$) whereas the V-nodes correspond to the columns of the matrix $\mathbf{H}$ (e.g. V_1 corresponds to column 1). If the corresponding row of a check node in the $\mathbf{H}$ matrix has a 1 at column v it means that this C-node is connected to the variable node V_v and vice versa. The degree of a node in the Tanner graph is the number of edges connected to this node (e.g. node V_8 has degree 4, C_3

has degree 5 in Fig. 1). One possibility to describe the structure of a group of LDPC codes, a so called code ensemble, is via the degree distributions

$$\lambda(x) = \sum_{i=2}^{d_l} \lambda_i x^{i-1} \tag{2}$$

and

$$\rho(x) = \sum_{j=2}^{d_r} \rho_i x^{j-1}, \tag{3}$$

with the maximum degree of variable nodes d_l and the maximum degree of check nodes d_r, respectively. In these distributions λ_i and ρ_i are the fractions of edges belonging to degree i variable and degree j check nodes, respectively [5].

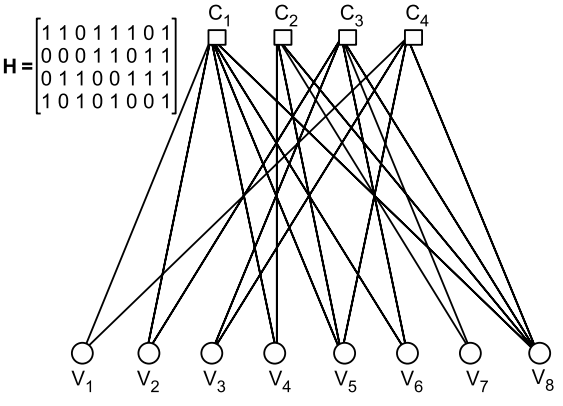


Fig. 1. Parity check matrix **H** and corresponding Tanner graph

For decoding of LDPC codes usually a message passing (or belief propagation) algorithm is used. In this algorithm messages are transmitted from variable nodes to check nodes and vice versa, via the edges of the Tanner graph. For additional information on the message passing algorithm see e.g. [6]. A common tool for analysis of this decoding method is *density evolution* [8]. In this work, density evolution is used to optimize the mapping of bits to modulation symbols.

4 Mapping of Bits to Modulation Symbols

The main purpose of this work is to use properties of the code structure to improve the performance of a communication system. As discussed before, the bits within one information symbol have different error probabilities. The idea is to map bits of a codeword with certain *variable node degrees* to specific bit positions of a modulation symbol.

4.1 Implementation of the Mapping

The mapping can be done very easily in hardware or software. When no special assignment is considered the bits of a data block to be transmitted are usually grouped into tuples of K bits and then assigned to symbols via a gray mapping. A different mapping can be done by a permutation of the bits before such a bit-to-symbol assignment. The permutation pattern defines the mapping of the bits to the modulation symbols. This leads to a structure of a communication system as shown in Fig. 2. Additionally, before the LDPC decoder, a depermutation has to be done. The gray box in Fig. 2 marks the components which are added for the proposed bit to symbol mapping. To perform an optimization of the

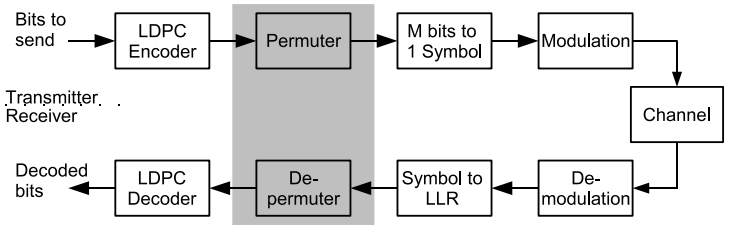


Fig. 2. Structure of communication system including the proposed mapping

bit-to-symbol mapping the follow mathematical model was derived: The combination of symbol mapping, AWGN channel and symbol demapping was modeled as multiple parallel AWGN channels using *BPSK modulation* for every channel. The simplified model uses *parallel virtual (BPSK) AWGN channels*, one for each bit of a modulation symbol. By using the bit error probability of bit k, the (linear) SNR value of the corresponding virtual AWGN channel using BPSK modulation can be calculated as [7]

$$\mathrm{SNR}_k = 2\ (\mathrm{erfc}^{-1}(2\ P_{bk}))^2. \tag{4}$$

In this simplification, the symbol mapping, the AWGN channel and the symbol demapping are simulated by these parallel AWGN channels. This leads to the same *uncoded* bit error probability as the combination of symbol mapping, AWGN and demapping. Although, because of statistical dependencies of the log likelihood ratios of the bits of a modulation symbol, the *coded* bit error rates (BER) of the simplified model are not the same as of the real simulation structure. But simulations showed that the principal characteristics discussed in this paper coincide for both simulation models.

To describe the mapping of bits of a codeword with a certain variable node degree to bit positions of a modulation symbol a mapping matrix $\mathbf{M}$ is defined. This matrix $\mathbf{M}$ of dimension $d_l \times K$ assigns which fraction of variable nodes with a specific degree are mapped to a virtual channel. Every row i of this matrix corresponds to the variable nodes of degree i, every column v to the virtual

channel v. The entry m_{iv} defines the fraction of variable nodes which have degree i and which are assigned to channel v. For example a mapping matrix for the code example shown in Fig. 1 which maps the bits with a highest variable node degrees to the most reliable bit positions in a modulation symbol looks like

$$\mathbf{M} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.3333 & 0.3333 & 0.3334 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}. \tag{5}$$

Please note that the values m_{iv} have to be set to values which are valid according to the used code ensemble (e.g. that the assignment of variable nodes of a specific degree does not exceed the number of availiable variable nodes of this degree).

4.2 Density Evolution

As a tool to compare the different symbol mapping schemes and to perform the optimization of the mapping, we used density evolution. This technique allows to monitor the evolution of probability densities of LLR values at the check and variable nodes of an LDPC ensemble. For simplicity Gaussian approximation was applied [8]. With this approximation the densities of the LLR values are assumed to be Gaussian, so only the mean (the variance can be easily derived from this mean) of this Gaussian distribution has to be tracked. According to [8] this mean $m_u^{(l)}$ can be calculated iteratively by

$$m_u^{(l)} = \sum_{j=2}^{d_r} \rho_j \phi^{-1}(1 - [1 - \sum_{i=2}^{d_l} \lambda_i \phi(m_{u_0} + (i-1)m_u^{(l-1)})]^{j-1}), \tag{6}$$

with l as the iteration number, d_r and d_l as the maximum check node and variable node degree, respectively, $\phi(x) = 1 - E[\tanh(x/2)]$ and m_{u_0} as the mean of the LLR values out of the channel (in [8] also simple approximation functions for $\phi(x)$ are presented). For an AWGN channel with zero mean and variance σ_n^2 this (LLR-)mean can be calculated as $m_{u_0} = 2/\sigma_n^2$.

In our work we investigated the impact of the assignment of bits corresponding to variable nodes of specific degrees to different bits positions of a modulation symbol with different reliabilities. So Eqn. 6 has to be modified for different means of the LLR values according to those bit positions. This can be accomplished by introducing the channel mean distribution

$$\mu(x) = \sum_{i=2}^{d_l} \mu_i x^{i-1}. \tag{7}$$

Here every coefficient μ_i specifies the Gaussian mean value of the variable node with degree i. With this, Eqn. 6 can be changed to

$$m_u^{(l)} = \sum_{j=2}^{d_r} \rho_j \phi^{-1}(1 - [1 - \sum_{i=2}^{d_l} \lambda_i \phi(\mu_i + (i-1)m_u^{(l-1)})]^{j-1}). \tag{8}$$

Now with this modification, density evolution is capable to take into consideration the different SNR values of the virtual AWGN channels, respectively the different bit positions. For application of density evolution, an allocation of the bits to the virtual channels has to be done. Every virtual channel v is characterized by its variance σ_{nv}^2 or equivalently by the mean m_{u_v} of the LLR values out of this channel. These mean values can be combined into the vector

$$\mathbf{m_u} = \left[m_{u_1}\ m_{u_2}\ \ldots\ m_{u_K}\right]^T. \tag{9}$$

With this vector and the mapping Matrix $\mathbf{M}$ the coefficients of the channel mean distribution can be calculated as

$$\boldsymbol{\mu} = \left[\mu_1\ \mu_2\ \ldots\ \mu_{dl}\right]^T = \mathbf{M}\mathbf{m_u}. \tag{10}$$

4.3 Optimization of Mapping

With the modified density evolution described in the previous section, the mapping schemes were optimized. This was done by searching those mapping matrix where density evolution converged to the highest mean value at a given noise level, using non-linear programming. Fig. 3 shows the density evolution results for optimized mapping schemes (curve (d)). As an example a code proposed for WLAN [2] was used. For each point of this curve, an optimal mapping matrix was calculated. Unfortunately, for the considered codes, no single mapping matrix performs best for all channel SNR values. So, for practical application, a mapping matrix has to be choosen according to the operating point of the communication system. The optimization led to the following rule of thumb: if the operating point of the communication system is at a high noise level then bits with a high degree should be mapped to reliable bit positions (Fig. 3 curve (a)), if the noise level is low then bits with a low degree should be mapped to the most reliable bit positions (Fig. 3 curve (c); curve (b) shows a random mapping for performance comparison).

5 Simulation Results

The prosed methods were evaluated by simulation. Again the WLAN code with rate 1/2 and a code length of 648 bits and 16-QAM modulation was used. Similar to the last section the curve (a) of Fig. 4 show the simulation results for the mapping of bits with a high degree to the more reliable bit positions, the curves (b) an arbitrary (random) mapping and the curves (c) the mapping of bits with a low degree to the more reliable bit positions. As shown in Fig. 4 the results of density evolution are confirmed by the simulation. Please note that because of the characteristics of density evolution (with assumtions like code word lengths of ∞, code ensembles instead of specific codes, ...) and statistical dependencies of the bits of a modulation symbol, only a qualitative comparison of different mapping schemes is possible with density evolution. Because of this, the scales of Fig. 3 and Fig. 4 are not comparable.

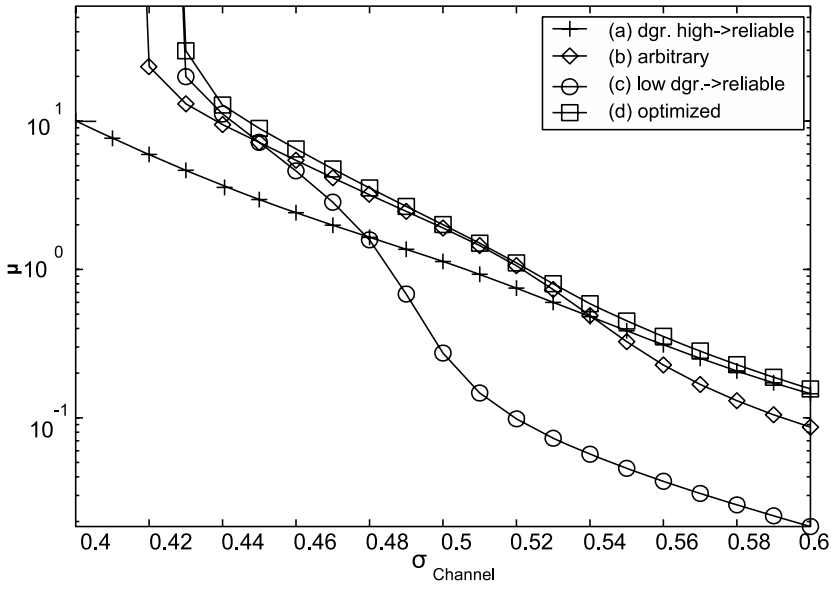


Fig. 3. Density evolution for degr. dist. of WLAN matrix 648 bits, rate 1/2.

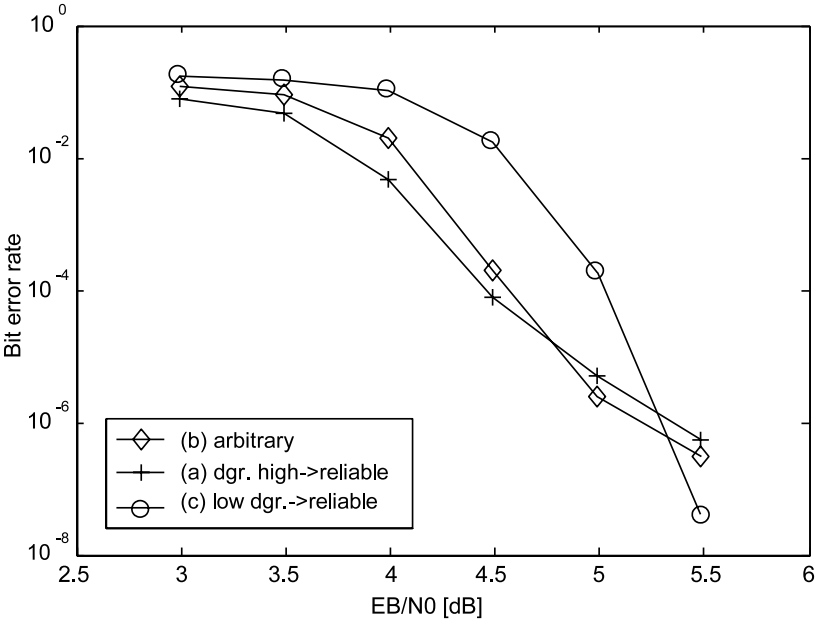


Fig. 4. Simulation results for WLAN matrix r=1/2, n=648 bits

6 Conclusion

In this work we applied theoretical insights, recently discovered about LDPC codes, to improve architectures for communication systems. We present how the state-of-the-art analysis tool for LDPC code ensembles, density evolution, can be modified to optimize the performance of a communication system when a specific code has to be used. Our simulation results confirmed, additionally to the theoretical analysis, that for low SNR values, or equivalently for regions with a high bit error rate, the best performance is achieved when the bits with the highest variable node degrees are mapped to the most reliable bits. This is especially important because many communication systems have its operating points at high noise levels.

References

1. Gallager, R.G.: Low-Density Parity-Check Codes. MIT Press, Cambridge, MA (1963)
2. Joint Proposal: High throughput extension to the 802.11 Standard: PHY. IEEE 802.11-05/1102r4 (January 2006)
3. Mannoni, V., Declercq, D., Gelle, G.: Optimized Irregular Low-Density Parity-Check Codes for Multicarrier Modulations over Frequency-Selective Channels. EURASIP Journal on Applied Signal Processing, 1546–1556 (2004)
4. Yoon, D., Cho, K.: General bit error probability of rectangular quadrature amplitude modulation. IEEE Electronics Letters 38(3), 131–133 (2002)
5. Luby, M., Mitzenmacher, M., Shokrollahi, A., Spielman, D., Stemann, V.: Practical loss-resilient codes. Proc. 29th Annu. ACM Symp. Theory of Computing, 150–159 (1997)
6. Moon, T.K.: Error Correcting Coding: Mathematical Methods and Algorithms. Wiley, New York (2005)
7. Proakis, J.G.: Digital Communications, 2nd edn. McGrawHill, New York (1989)
8. Chung, S.Y., Richardson, T., Urbanke, R.: Analysis of sum-product decoding of low density parity check codes using Gaussian approximation. IEEE Transactions on Information Theory 47, 657–670 (2001)

Texture-Based Filtering and Front-Propagation Techniques for the Segmentation of Ultrasound Images

Miguel Alemán-Flores[1], Patricia Alemán-Flores[2], Luis Álvarez-León[1], M. Belén Esteban-Sánchez[1], Rafael Fuentes-Pavón[2], and José M. Santana-Montesdeoca[2,⋆]

[1] Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria, 35017, Las Palmas, Spain
[2] Sección de Ecografía, Servicio de Radiodiagnóstico
Hospital Universitario Insular de Gran Canaria, 35016, Las Palmas, Spain
besteban@dis.ulpgc.es

Abstract. Ultrasound imaging segmentation is a common method used to help in the diagnosis in multiple medical disciplines. This medical image modality is particularly difficult to segment and analyze since the quality of the images is relatively low, because of the presence of speckle noise. In this paper we present a set of techniques, based on texture findings, to increase the quality of the images. We characterize the ultrasound image texture by a vector of responses to a set of Gabor filters. Also, we combine front-propagation and active contours segmentation methods to achieve a fast accurate segmentation with the minimal expert intervention.

1 Introduction

Ultrasound (US) imaging has become an important diagnostic tool in medicine [1]. Mainly, this medical image modality is known by its application in gynecology but US imaging is also important for diagnosis of abdominal organs or heart, fetal examination, etc. Its main advantages are that it is a cost effective, quick, painless and non-invasive technique. Among the disadvantages, we found that US imaging needs expert interpretation and the presence of a special kind of noise, called speckle.

Ultrasound plays a crucial role in the diagnosis of breast cancer. Distinguishing benign and malignant nodules in breast US images is a useful way to avoid unnecessary biopsies. Most of the diagnostic criteria require an accurate segmentation of the nodule inside the image [2]. Manual segmentation performed by the radiologist is an expensive and time-consuming task that can be carried out with computer assisted methods. On the other hand, fully automatic techniques for ultrasound segmentation are not robust. We present a semi-automatic method

⋆ Alphabetical order has been used in the author's list.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 960–967, 2007.
© Springer-Verlag Berlin Heidelberg 2007

to extract nodule contours. It combines some segmentation methods with the minimal intervention of the radiologist, that selects a point inside the nodule.

The presence of speckle noise in ultrasound images degrades the visibility of diagnostic criteria (nodule shape and contour). It is necessary to reduce it before processing the image. Many methods have been proposed, including the truncated median filter [3], anisotropic diffusion [4] and speckle-reducing anisotropic diffusion [5]. Most of them do not preserve key and useful details, such as edges. We propose to preprocess the image, applying a variation of the classical anisotropic diffusion scheme, guiding it by Gabor texture descriptors.

The paper is structured as follows: Section 2 presents the texture-based filtering technique and focuses on Gabor filter as a texture feature extractor. Section 3 explains the use and combination of the front-propagation and active contours methods to segment regions of the ultrasound images. The work is concluded with an account of our main conclusions (Section 4).

2 Texture-Based Ultrasound Filtering

Before processing US images, it is necessary to reduce the presence of speckle noise. The aim is to remove it without blurring or distorting edges that could affect the diagnostic details.

The speckle noise in ultrasound images is generally modelled as a multiplicative noise. Classical image filters, such as gaussian are focused on removal of additive noise, so they are not suitable for speckle filtering. Other common techniques include median filter and anisotropic diffusion filters.

2.1 Anisotropic Diffusion

A good result for US images is obtained by applying the anisotropic diffusion, introduced by Perona and Malik [4]. This method filters the image, smoothing inside the different regions in preference to smooth across the boundaries:

$$\begin{cases} \frac{\delta I}{\delta t} = div(g(\|\nabla I\|)\nabla I) \\ \\ I(t=0) = I_o \end{cases} \tag{1}$$

where I_o is the original noisy image. The gradient magnitude $\|\nabla I\|$ is used to detect the edges, that stop the diffusion through the function $g(.)$

$$g(x) = e^{-\alpha x^2} \tag{2}$$

α is a contrast noise estimator, computed from the gradient magnitude histogram [6].

Gradient magnitude is not a good border detector in US, because of the presence of speckle. The texture contains information about the different areas or tissues in the image. Consequently, to improve the Perona-Malik scheme, we propose to include the texture information as a guide for the diffusion process.

2.2 Texture Extraction Through a Gabor Filter Bank

Texture is defined as a statistical pattern of the grey levels in the image. It is not specified by the intensity in a single point, it is always based on some neighbourhood. We extract texture features from the grey value texture pattern in the ultrasound images. The texture is characterized by a vector of scalar descriptors r_i, computed as the responses of an image I to different filters

$$R = \{r_1, ..., r_n\} \tag{3}$$

Bidimensional Gabor filters [7] are commonly used for image texture extraction [8] [9] [10] [11] and segmentation of US images [12] [13]. They represent a generalization of the functions proposed by Gabor [14], and model the response of some mammalian visual cortex cells [15]. They are the product of a gaussian by a complex sinusoidal wave

$$g(x,y) = e^{(-\pi(\frac{x_\omega^2}{\sigma_x^2}+\frac{y_\omega^2}{\sigma_y^2}))} e^{(i(2\pi\lambda(x\cos\omega+y\sin\omega)))} \tag{4}$$

where

$$\begin{cases} x_\omega = (x-x_0)\cos\omega + (y-y_0)\sin\omega \\ y_\omega = -(x-x_0)\sin\omega + (y-y_0)\cos\omega \end{cases} \tag{5}$$

σ_xe σ_y are the gaussian standard deviations. (x_0, y_0) and (λ, ω) set the gaussian position in the spatial and frequency domain respectively. This function is a bandpass filter, associated to a specific range of frequencies (or filter band). It responds to signals oriented in a specific orientation (or filter direction). The real asymmetrical part acts as a smooth filter, while the imaginary symmetrical part acts as a border detector. Figure 1 shows the real and imaginary part of a Gabor filter in the spatial domain.

To compute the texture vector, we filter the image with a set of Gabor filters, called Gabor filter bank. The filters have different values for the parameters σ_x, σ_y, λ ,ω. A typical design strategy consist in avoiding the overlap of the half-magnitude filter responses in the frequency domain, while covering the whole domain. All the filters have the same half-magnitude frequency bandwidth in octaves. Figure 2 shows a Gabor filter bank with 6 bands and 2 orientations.

Using a filter bank of N different bands and M different orientations, we obtain a NxM responses F_n^m, with $1 \leq m \leq M$ and $1 \leq N \leq N$, that are the components of the texture vector R. The responses contain information about orientation and frequencies of the intensity patterns in the neighbourhood of each point.

2.3 Texture Guided Anisotropic Diffusion

We have modified the anisotropic diffusion scheme by changing the intensity gradient term by the gradient of the texture vector

$$\frac{\delta I}{\delta t} = div(g(\|\nabla R\|)\nabla I) \tag{6}$$

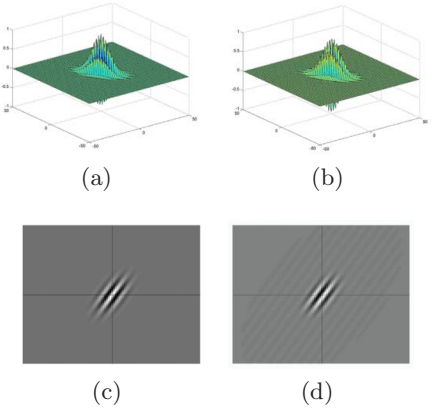


Fig. 1. A Gabor filter in the spatial domain $\sigma_x = 10$, $\sigma_y = 20$, $\lambda = 0.20$, $\theta = \pi/3$, $x_0 = y_0 = 0$, $\gamma = 0$. a) Real part. b) Imaginary part. c) Real part magnitude. d) Imaginary part magnitude.

The numerical implementation of this equation is as follows. We compute the gradient magnitude of the texture vector for a pair of pixels p,q of the image as

$$\|\nabla R(p,q)\| = \sqrt{\sum_{1\leq m\leq M} \sum_{1\leq n\leq N} (F_n^m(p) - F_n^m(q))} \tag{7}$$

At each iteration of the diffusion, the new pixel value $I_p^{t+\Delta t}$ is computed as

$$I_p^{t+\Delta t} = I_p^t + \frac{\Delta t}{\|\eta\|} \sum_{q\in\eta} g(\|\nabla R(p,q)\|)\nabla I_{p,q}^t \tag{8}$$

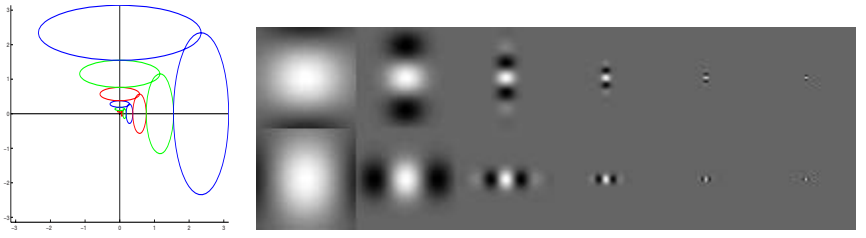
where η represents the neighbourhood of pixel p.

Figure 3 shows a breast ultrasound image and the smoothed version obtained by the truncated median filter, the classical anisotropic diffusion and the texture-based anisotropic diffusion.

3 Front-Propagation Segmentation of Ultrasound Images

Some accurate segmentation techniques, such as active contours methods [16], need the initialization of the curve. This time-consuming task is not always available. Some other methods, such as region-growing [17] can obtain a rough draft of the nodule contour setting only one point. However, this implementation needs a proper threshold to distinguish the inner region of the nodule.

We propose to combine front-propagation segmentation [18] and active contours to obtain a segmentation of the nodule.



(a) (b)

Fig. 2. Gabor filter bank of 6 bands and 2 orientations. a) Half peak magnitudes of the filters. b) Real part magnitudes in the spatial domain.

3.1 Front-Propagation Pre-segmentation

The radiologist can easily select a point inside the nodule. This initial point is expanded outside with the front propagation technique, until the gradient magnitude is high enough to stop the process (near the edges)

$$\begin{cases} \frac{\delta u}{\delta t} = -g(\|\nabla I_s\|)\nabla I_s \\ \\ u(t=0) = u_0 \end{cases} \tag{9}$$

where u represents the level set, I_s is the filtered image and u_0 the initial point. The presence of an edge stops the process through the function $g(.)$

$$g(I_s) = \frac{1}{\sqrt{1+\alpha\|\nabla I_s\|^2}} \tag{10}$$

With this method, we obtain a rough segmentation of the nodule, that is improved by the active contours, also called snakes.

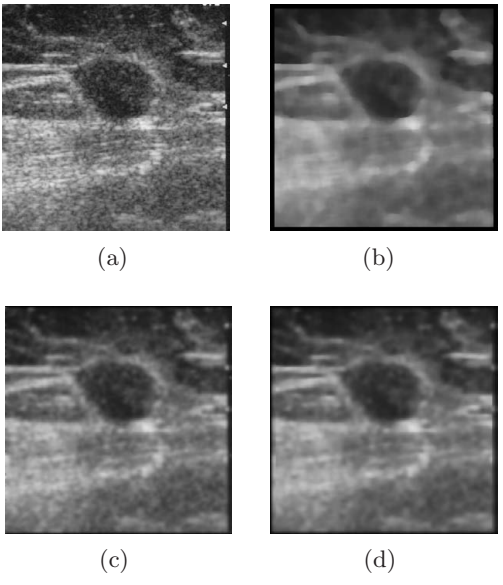
3.2 Active Contours Segmentation

This method consists in forth deforming an initial contour of the object to find the object boundaries, following a set of internal and external forces. We use the following scheme

$$\begin{cases} \frac{\delta v}{\delta t} = g(I_s)\|\nabla v\| div(\frac{\nabla v}{\|\nabla v\|}) + \lambda \nabla v \nabla g(I_s) \\ \\ v(t=0) = ut = n \end{cases} \tag{11}$$

where v represents the snake and $v_{t=0}$ is the front-propagation segmentation. λ is a parameter that regulates the attraction force.

Figure 4 shows a comparison between manual delimitation of the nodule in image in 3, and region-growing and front-propagation semi-automatic segmentations.



(a) (b)

(c) (d)

Fig. 3. Breast ultrasound image filtered with different smoothing methods. a) Original US image. b) Truncated Median Filter. c) Anisotropic Diffusion. e) Texture-based Anisotropic Diffusion.

3.3 Numerical Results

To test the method, we use two different measures. We have two segmentations: the manual one performed by the radiologist (A) and the semiautomatic one (B).

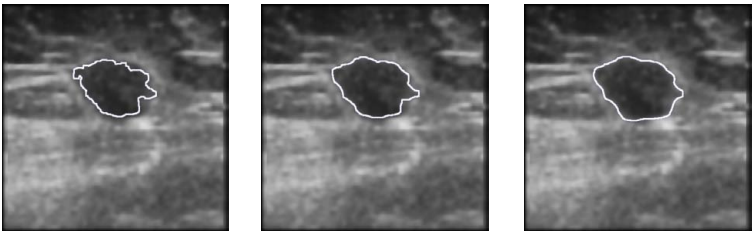
The first measure is the coincidence percentage: it is defined as the area of the intersection of the two segmentations divided by the area of the union and multiplied by 100. When two segmentations are identical, this coincidence percentage is 100.

$$CP(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|}.100 \tag{12}$$

The second measure is the proportional distance. We extract the contour of both segmentations: C_1 and C_2. We compute the distance from each point of the first contour to the other and vice-versa. The sum of these quantities is divided by the square root of the area of the manual segmentation and corrected by a scale factor. When two segmentations are identical, this proportional distance is close to 0.

$$PD(A,B) = \frac{\frac{\sum_{x_i \in C_1} d(x_i,C_2)}{|C_1|} + \frac{\sum_{x_i \in C_2} d(x_i,C_1)}{|C_2|}}{2\sqrt{|S|}}.\sqrt{\pi}.100 \tag{13}$$

We have tested the method in a set of 30 breast US images. The performance of our approach is presented in table 1. The best results are obtained with the combination of front propagation and active contours.



(a) (b) (c)

Fig. 4. Comparison of different segmentation methods in a breast US image: a) Region-growing segmentation. b) Front-propagation segmentation. b) Combination of front-propagation and active contours segmentation.

Table 1. Numerical results for different segmentation methods in a set of 30 US breast images

Segmentation Method	PD	CP
Region-growing	9.70	76.68
Front-propagation	8.01	81.74
Front-propagation + Active Contours	5.84	86.94

4 Conclusions

We introduce a method that uses the image texture information to filter the speckle noise present in the US images. The modification of the Anisotropic diffusion scheme, adding the texture information, increases the results of this method in the US images. The texture is computed through the Gabor filter bank responses.

We propose the combination of front-propagation and active contour (snakes) methods to obtain the shape of the nodule. This method allows obtaining a widely precise contour of an object inside an image. We use a rough pre-segmentation as initial snake. This pre-segmentation is obtained by a front propagation procedure. The initial seed of the algorithm is set by the radiologist, who selects an internal point of the nodule. Our methods offer higher accuracy, compared with other approaches.

References

1. Erikson, K.R., Fry, F.J., Jones, J.P.: Ultrasound in medicine-a review. IEEE Transactions on Sonics and Ultrasonics 21(3), 144–170 (1974)
2. Stavros, A.T., Thickman, D., Rapp, C.L., Dennis, M.A., Parker, S.H., Sisney, G.A.: Solid breast nodules: use of sonography to distinguish between benign and malignant lesions. Radiology 196, 123–134 (1995)

3. Davies, E.R.: On the noise suppression and image enhancement characteristics of the median, truncated median and mode filters. Pattern Recogn. Lett. 7(2), 87–97 (1988)
4. Perona, P., Malik, J.: Scale space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)
5. Yu, Y., Acton, S.T.: Speckle reducing anisotropic diffusion. IEEE Transactions on Image Processing 11(11), 1260–1270 (2002)
6. Voci, F., Eiho, S., Sugimoto, N., Sekibuchi, H.: Estimating the gradient in the perona-malik equation. IEEE Signal Processing Magazine 21(3), 39–65 (2004)
7. Daugman, J.G.: Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. IEEE Transactions on Acoustics, Speech and Signal Processing 36(7), 1169–1179 (1988)
8. Bovik, A.C., Clark, M., Geisler, W.S.: Multichannel texture analysis using localized spatial filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(1), 55–73 (1990)
9. Weldon, T.P., Higgins, W.E.: Design of Multiple Gabor Filters for Texture Segmentation. In: vol. 4, pp. 2243–2246 (1996)
10. Dunn, D., Higgings, W.E.: Optimal gabor filters for texture segmentation. IEEE Transactions on Image Processing 4(7), 947–964 (1995)
11. Anil, K., Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using gabor filters. Pattern Recogn. 24(12), 1167–1186 (1990)
12. Mohamed, S.S., Abdel-galil, T.K., Salama, M.M.A., Fenster, A., Rizkalla, K., Downey, D.B.: Prostate cancer diagnosis based on gabor filter texture segmentation of ultrasound image. In: CCECE 2003, vol. 3, pp. 1485–1488 (2003)
13. Xie, J., Jiang, Y., Hung-Tat, T.: Segmentation of kidney from ultrasound images based on texture and shape priors. IEEE transactions on medical imaging 24, 45–57 (2005)
14. Gabor, D.: Theory of communication. Journ. of Inst. Electrical Engineers 93(26), 429–457 (1946)
15. De Valois, R.L., De Valois, K.K.: Spatial Vision. Oxford University Press, Oxford (1988)
16. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: ICCV, pp. 694–699 (1995)
17. Sato, M., Lakare, S., Wan, M., Kaufman, A., Nakajima, M.: A gradient magnitude based region growing algorithm for accurate segmentation. In: Proceedings of the International Conference on Image Processing, vol. 3, pp. 448–451 (2000)
18. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. J. Comput. Phys. 79(1), 12–49 (1988)

"Chameleon" Software Defined Control Platform

Adam Handzlik[1] and Andrzej Jablonski[2]

[1] Microtech International Ltd Fabryczna 13 str., 53-609 Wroclaw, Poland
a.handzlik@microtech.com.pl
http://www.microtech.com.pl
[2] Faculty of Electronics the Wroclaw University of Technology
27 Wybrzeze Wyspianskiego str., 50-370 Wroclaw, Poland
andrzej.jablonski@pwr.wroc.pl
http://eka.pwr.wroc.pl

Abstract. This paper presents a basic concept, development and implementation of software tool designated for programming of Field Programmable Gate Arrays. However, the programming scheme does not employ typical VHDL or Verilog languages but dedicated PLC-like software environment producing both execution code and hardware configuration for the for FPGA chip. As a result the project has created PLC language defined control system implemented in an FPGA structure.

Verification tests were performed on Xilinx Spartan II platform and finally were converted into Spartan III chip family. Although verification tests were run using development boards the real life embedded application is now developed.

Keywords: Signal processing architectures, control platform development, innovative reprogrammable technology, Field Programmable Gate Arrays, IP Core, virtual Programmable Logic Controller.

1 Introduction

Nowadays control hardware is installed in almost all machines and equipment, used in different domains of life. Concept of electronic control makes our life easier and safer in various meanings. Many industry manufacturers in Europe develop and manufacture hardware control platforms every day. They have to be still in motion by improving their control systems to be competitive on European and world markets, always a step ahead towards their rivals.

This paper presents a new Software Suite for control platform development and implementation. Features of that tools allow controller manufacturers radically reduce the time spend on hardware configuration development, using the latest new innovative reprogrammable technology, FPGA (Field Programmable Gate Arrays). Today, this technology is highly efficient in control solutions and makes controllers faster and more flexible, but is very hard to program and implement. Our software tool dramatically improve the programming and implementation of this technology. It makes European manufacturers more competitive

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 968–975, 2007.
© Springer-Verlag Berlin Heidelberg 2007

in continuing contest on markets by increasing their efficiency and productivity, reducing manufacturing costs and enlarging quality of their products. It also provides the capability to manage the entire system in a collaborative way, improving the flexibility and reducing the entire cost of management of the project. Finally, it significantly contributes to PLM (product life-cycle management).

2 Project

The main driver of the project was to develop a programming environment and standard IP Core modules, which allow users building their own controller architecture with various processing engines, I/Os, communication ports, etc., in a simple way - just by choosing proper blocks and inserting them into an application. One of these blocks is a real-time Virtual Machine replacing standard semi-code interpreter to allow creation of control application in a simple way - using standard PLC programming tools. The application made by user is the source program, operating on standard FPGA structures.

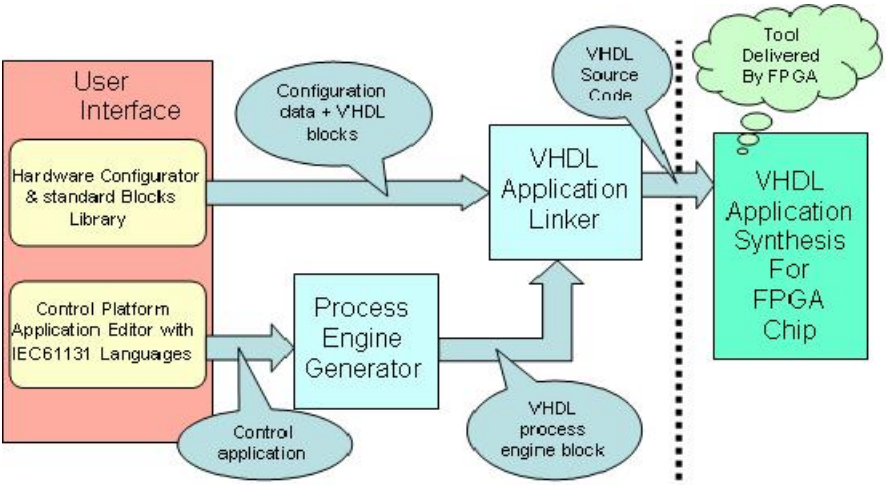


Fig. 1. Block diagram of Chameleon programming environment

It radically simplifies the use of FPGA technology in control hardware development. Actually FPGA technology is not available for many manufacturers because of complicated manner of FPGA matrix programming. Proposed solution allows for commercial employment of FPGA technology and for taking benefits from FPGA advantages by many large and small manufacturers on the European market. Moreover, different control platforms (features depend on selected application) can be implemented in the same standard hardware board (!),which makes manufacturing of control solution much cheaper and faster.

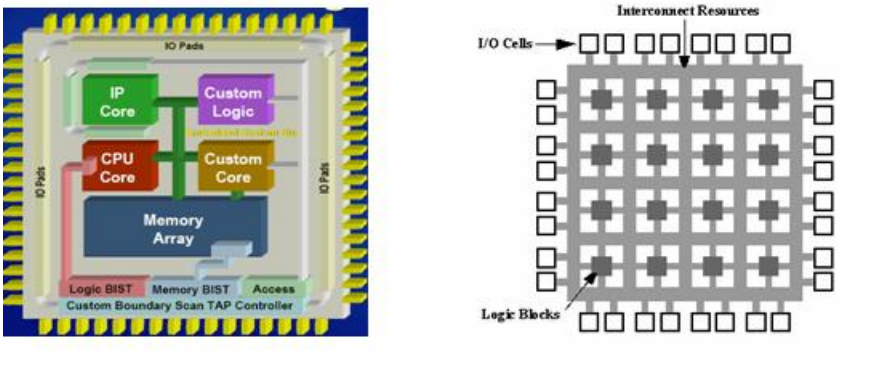


Fig. 2. Field Programming Gates Array (FPGA) structure

As mentioned above, control system is defined by composing blocks on high programming level. This new attitude to control system construction brings several benefits by:

- saving time & cost of development for OEMs
- maximizing hardware efficiency
- allowing true reusability of IP Cores

The solution features include:

- integrated and distributed software components
- easy,user friendly environment for control hardware design
- libraries with ready to use block components; library can be completed with new blocks
- web-based, distributed application
- collaborative application (enables exchange of information and sharing of responsibilities between co-workers and all part of a system).

2.1 Generic Virtual Machine Structure

Generic Virtual Machine is built of units, databases, several type of junctions and two types of blocks: executive block UNIT and data storing block DATA. Each block share Address, Data and Control Buses. All units are masters while all data blocks are slaves on the bus interconnecting units and busses.

Units can execute fragments of program code or perform operations on variables. Program flow controlling unit prepares arguments used by units and coordinates the program execution.

Only one unit can be active at once. As long as a unit is active, it puts its token onto the token line of Unit Interconnection Bus. When the unit completes its task it puts the number of calling unit back onto the token line together with optional return value at PARAM line as shown in Fig. 4.

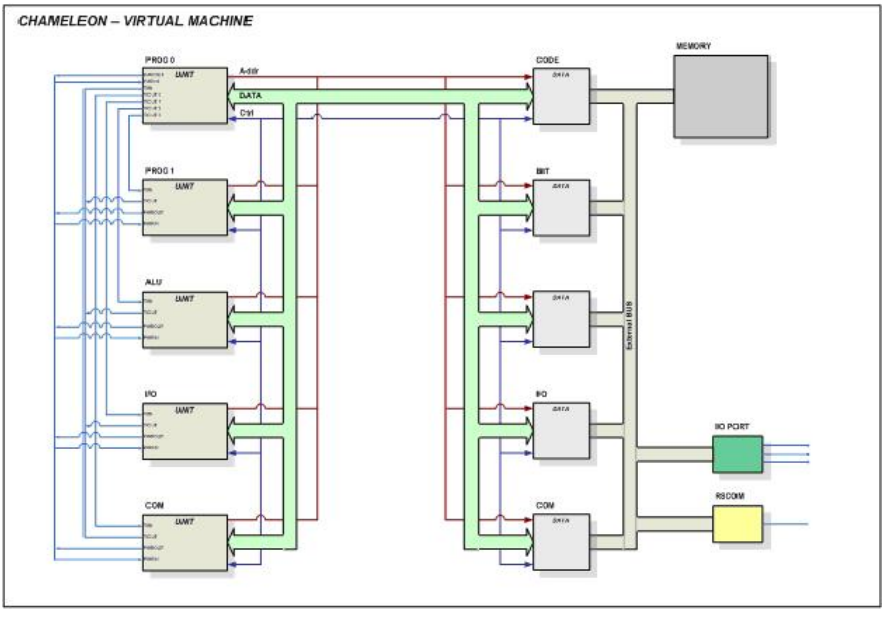


Fig. 3. Structure of the Virtual Machine

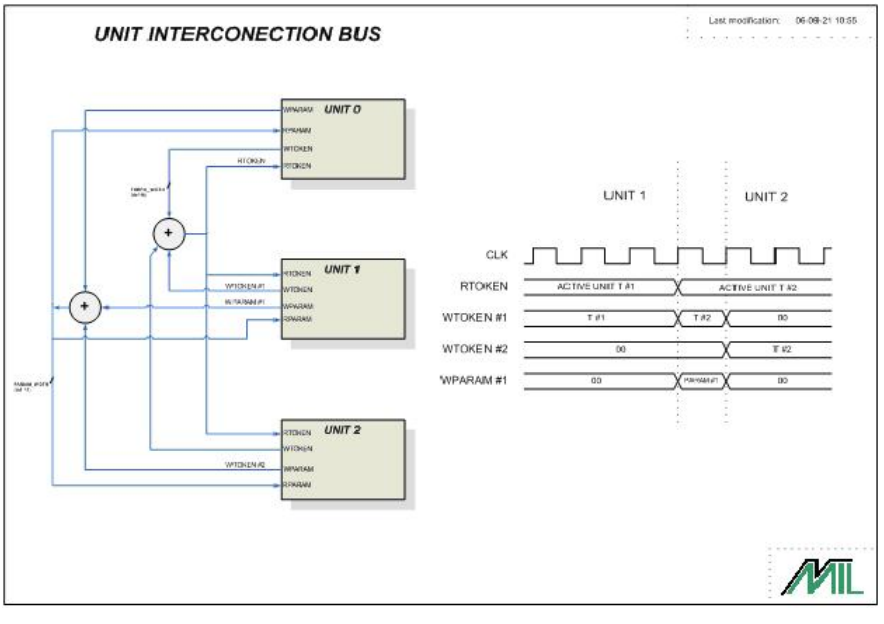


Fig. 4. Unit Interconnection Bus

In case of calling another unit, caller puts desired unit number onto the token line together with optional parameter on the PARAM line of the bus. Active unit has an exclusive control over the Virtual Machine.

The general structure of Virtual Machine of Chameleon system is presented in the Fig.3.

Data blocks link virtual machine variable space with internal registers, memories and external devices. All data in the variable space is indexed within the data types. Different data types are located in separate databases.

Every data block has registers updated by the program flow controller to store index of arguments of currently executed command. This approach reduces the need of passing addresses by the use of indirect addressing through the database registers. Function return value is stored through the register number 0; arguments of the function occupy locations above. Selection of data block thus the data type is controlled by the active unit.

2.2 Software Structure

The software part of the project is comprised of:

- the Project Wizard to choose the board and Virtual Machine template,
- the Chameleon Configuration Editor to edit the whole Virtual Machine Environment including SRAM, blocks linking external devices, I/O connections, bind physical FPGA I/Os

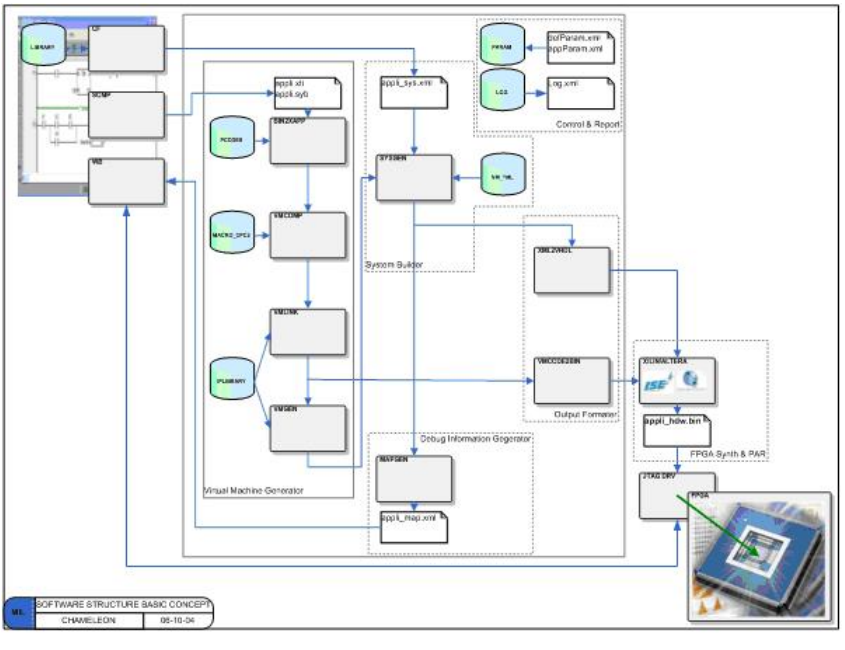


Fig. 5. Chameleon software structure - basic concept

- the PLC program code editor to write code in the following PLC programming languages:
  - Function Block Diagram (FBD)
  - Ladder Diagram (LD)
  - Structured Text (ST)
  - Instruction List (IL)
- the compiler, linker and VHDL file generator

The code edited in the development environment is translated into xml description of the Virtual Machine Core by the Virtual Machine Generator, system description.

Control system will be defined by composing blocks on high programming level and by writing application software for control in high-level languages corresponding to IEC 61131 standard. These innovations have a huge economic impact regarding the return on investment. It is actually possible to start a rapid system implementation and take advantage of FPGA technology, further expanding it towards a higher sophistication level. Our solution is therefore a functional and technical innovation.

3 Tools for Solution Development

We used advanced development tools for project realization, delivered by leading tool manufacturers including software creation and chip programming environments. One of this tools is Microsoft Visual Studio .Net.

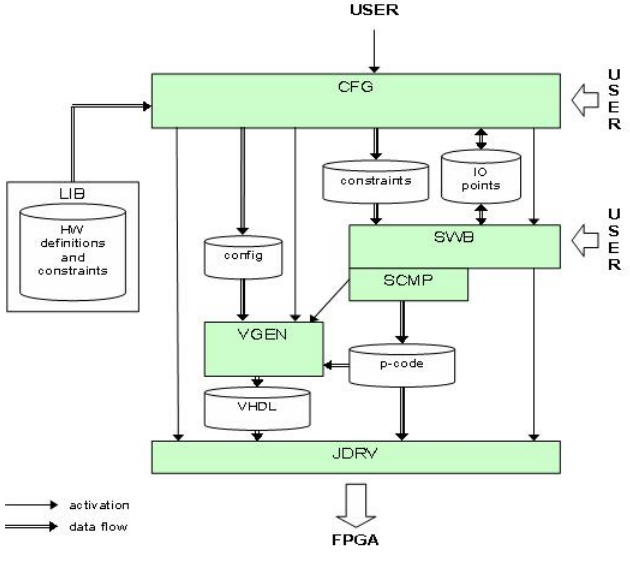


Fig. 6. Final structure of Software Defined Control Platform CHAMELEON

Microsoft.NET is a set of software technologies for connecting information, people, systems, and devices. This new generation of technology is based on Web services-small building-block applications that can connect to each other as well as to other, larger applications over the Internet. The "Chameleon" merges friendly programming interface (PLC based) and implementation of designed control application in hardware using FPGA technology. The application linker automatically produce VHDL code from predefined standard blocs and translate application program for VHDL Application Synthesis Tool.

Chameleon Software has many integrated useful tools for debagging, simulation and monitoring which make programming more convenient and reduce development time.

4 Development Platform

The Virtual Machine has been developed using MIL Spartan2 demoboard. The board consists of XC2S200 Spartan2 FPGA chip together with suitable power supplies, JTAG connector, and clock generator working at 18.432MHz, four 8-section switches, 8 pushbuttons and 32 light emitting diodes.



Fig. 7. MIL Spartan2 demoboard

5 Conclusions

The goal of Chameleon project is to create a configurable virtual Programmable Logic Controller (PLC) embedded in FPGA device with set of tools to enable programming of the controller using conventional PLC programming languages. It is designed as a tool for automation engineers who are not familiar with FPGA technique to let them program highly specialized logic controllers. The target customers will be European manufacturers of hardware control platform for different areas of applications, including manufacturers of: industrial controllers and machines, vehicles (automotive industry), medical equipment, computers, telecommunication equipment and many others.

Development of Software Defined Control Platform (SDCP) - acronym: "CHAMELEON"- shall give a new, innovative powerful software tools for European embedded control systems manufacturers and shall make them stronger, more flexible and competitive in our constantly changing market. The Chameleon Software was accomplished under patronage of the International Initiative Committee EUREKA! as Polish-French project "Software development project for easy control platform definition and implementation and common FPGA technology application for European control system manufacturers".

References

1. Pichler, F.: Walsh-Functions: Early ideas on their application in Signal Processing and Communication Engineering. In: Proceedings of The 2004 International TICSP Workshop on Spectral Methods and Multirate Signal Processing, SMMMSP 2004, ienna, Austria (September 11-12, 2004), pp. 263–269 (2004)
2. Pichler,F.: Mathematik als Kulturbeitrag - zum Andenken an Egmont Colerus. Kepler Symposium Philosophie und Geschichte der Mathematik. (Hrsg. MaaßLanger-Larcher). Vorträge aus dem Johannes Kepler Symposium 1995 bis 2005.(2005)
3. Pichler, F.: Der Computer - die multifunktionale Wundermaschine. Ausstellungskatalog des O Ö Landesmuseum Linz, Technik Ausstellung, in Vorbereitung (2006)
4. Keating, M., Bricaud, P.: Reuse methodology manual - for system-on-a-chip design. Kluwer Academic Publishers, Boston (2001)
5. Hsu, Y.-C., Tsai, K.F., Liu, J.T., Lin, E.S.: VHDL Modeling for Digital Design Synthesis. Kluwer Academic Publishers, Boston (1998)
6. Short, S.: BUILDING XML WEB SERVICES FOR MICROSOFT.NET PLATFORM. Polish edn. Microsoft Press, Warszawa (2003)
7. Microsoft Official Course: 2557A - Building COM+ Applications Using Microsoft.NET Enterprise Services (2002)
8. XILIX, QUALIS: FPGA Reuse Field Guide. Qualis Design Corporation (2000)
9. Skotarczyk, A., Englert, T., Chorazyczewski, A., Handzlik, A.: CHAMELEON-Software Development Project for Easy Control Platform Definition and Implementation and Common FPGA Technology Application for European Control System Manufacturers. Microtech International Ltd. Report, Wroclaw (2005)

Sampling Rate Conversion for Timing Adjustment in 3.5G Multimode Mobile Terminals

Dietmar Wenzel

Infineon Technologies AG, IFAG COM RF SE,
Am Campeon 1-12, 81726 Neubiberg, Germany
Dietmar.Wenzel@infineon.com

Abstract. Multimode Mobile Terminals require flexible signal processing architectures to support different mobile standards like GSM/EDGE and UMTS/HSDPA in one platform with minimum hardware resources. In this paper, an efficient architecture is presented using same system clock frequency for UMTS and GSM signal processing and an all-digital asynchronous sampling rate converter. In addition, it is shown how this architecture can be efficiently used to solve the problem of timing adjustments in time continuous signals of the UMTS/HSDPA receive and transmit path. The traditional and the proposed approaches are illustrated.

1 Introduction

Up to now most multimode mobile terminals have been using separate signal processing paths to support GSM and UMTS standard. Furthermore, RF transceiver signal processing and also the interfaces to baseband processors were mainly analog and separate for GSM/EDGE (2.5G) and UMTS/HSDPA (3.5G) signals. This led to a large number of analog interface wires, analog buffers and A/D or D/A converters. From now on, where new features like receive diversity using two independent receive paths even would need more interface connections, digital interfaces and optimized digital signal processing architectures become essential. This also allows employing digital measures for timing adjustment and reuse of functional blocks in different modes, even if the supported standards required nearly incommensurable symbol rates.

The goal of the approach proposed in this paper is of course exactly to solve this problem. The solution is achieved by employment of all-digital asynchronous sampling rate converters (SRC) in the 3.5G signal paths for transmit and receive direction. These sampling rate converters can efficiently be implemented on silicon for high bandwidth consuming services like 3.5G with low power suited for mobile applications.

2 The Clock and Timing Problem

On the one hand you have system given boundary conditions like:

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 976–983, 2007.
© Springer-Verlag Berlin Heidelberg 2007

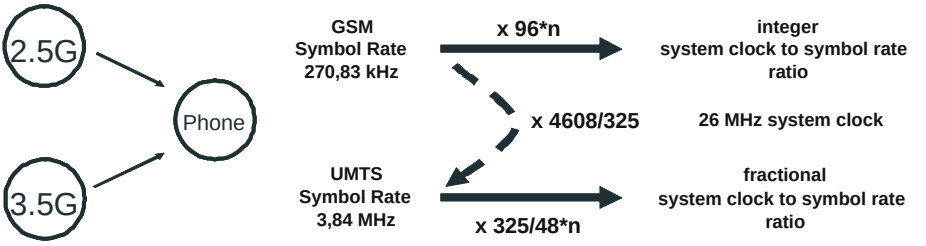


Fig. 1. Symbol and system clock rate ratios in a multimode terminal

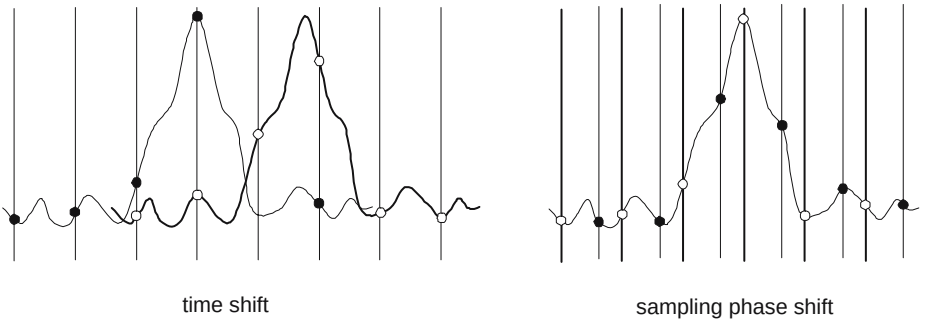


Fig. 2. Time and phase correction in a time-continuous signal

- Different mode dependent incommensurable symbol rates, e.g. 270,8$\overline{3}$ kHz for 2.5G and 3,84 MHz for 3.5G, which can not easily be derived from a single clock source (Fig. 1)
- The need for timing adjustment of time continuous signals in 3.5G if the signal content is not at wanted position in time (time shift) or is not correctly placed on sampling grid (phase shift) (Fig. 2).

On the other hand you have contrary implementation targets like:

- One system oscillator and one system clock frequency due to cost reasons
- Simple PLLs and non mode dependent constant operating frequencies
- Flexible selection of frequencies for DSP, controllers, signal processing hardware and architecture definition
- Reuse of blocks for different modes (2.5G and 3.5G)
- Constant number of symbols and samples per frame for baseband processing

These different requirements have to be met in a well-balanced way to achieve an optimum architecture.

Sampling rate conversion as well as time and phase shifts can be carried out by interpolation of samples with correct phase/time by means of digital signal processing.

Once sampling rate converters are part of a signal processing architecture they also can be used to solve those contrary tasks without increasing computational effort but giving a lot of flexibility in designing the remaining system architecture. For example, the system clock used to operate the baseband signal processing does not need to be an integer multiple of the standards symbol rate, it can be chosen to achieve maximum processing performance on the DSP or maximizing throughput in a data path.

3 Architectural Approach

The traditional solution in Fig. 3 uses two PLLs and generates individual system clocks for each subsystem. There is no common system clock on chip and in each mode is processing is done with an integer multiple clock frequency of the symbol rate. The timing adjustment in is carried out in 3.5G by repeating or omitting samples. Therefore, threre is no constant number of chips/samples per frame and scheduling is often more complicated. Futhermore, this requires comparatively high oversampling (8-10 times) and comparatively high memory requirements (receive buffer).

The innovative solution fulfills all requirements from Sec. 2 and additionally the same PLL can also be used for the digital RF-BB interface DigRF.

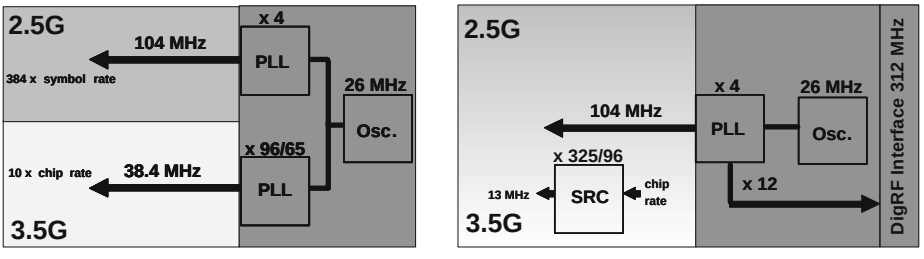


Fig. 3. Innovative solution saves one PLL and uses SRC

Fig. 4 shows an architecture based on the innovative solution from Fig. 3 for a multimode mobile terminal employing receive diversity.In the RF transceiver a lot of the signal processing resources are commonly used, i.e. the complete receive paths for 2.5G and 3.5G and parts of the transmit path. Only reconfiguration of various parameters is needed to switch between 2.5G and 3.5G. The SRCs perform the transition from the system defined sampling rates to the individual mode dependent symbol rates and therefore allow reuse of the receive chain for both modes. Hence, the SRCs in the receive paths can be either configured to output Tc/2 spaced samples with 7,68 MHz for 3.5G or T/2 spaced samples with $541,\overline{6}$ kHz for 2.5G while the physical operating frequency of the hardware remains constant.

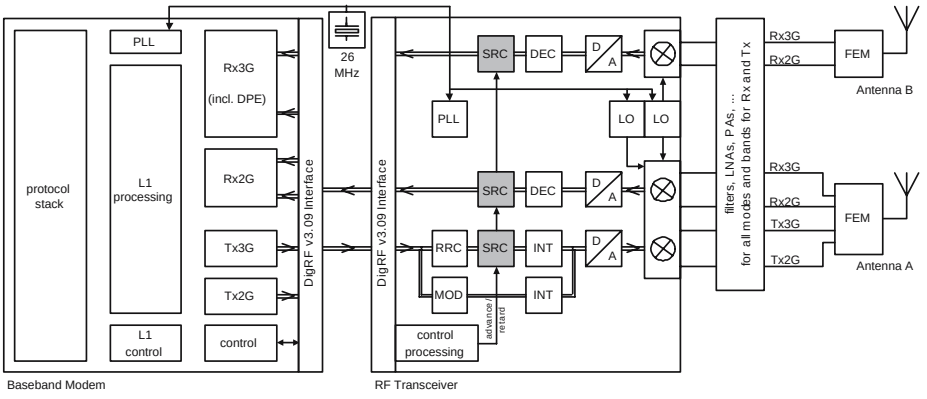


Fig. 4. Multimode mobile terminal with receive diversity using a DigRF v3.09 digital interface and sampling rate converters (SRC) employing one fixed system oscillator and fixed operating frequencies

It also shall be mentioned that there are other possible approaches where asynchronous sampling rate conversion could be applied in the 2.5G path. However, due to evolutionary reasons most mobile terminals use system clock frequencies which are derived from the GSM world and are based on a 26 MHz system oscillator. Therefore, the newer 3.5G system often has to adapt to these boundary conditions in a multimode mobile terminal.

4 Timing Correction in a Time-Continuous Signal

Another benefit of asynchronous SRCs is their ability of dynamically adjusting the converting ratio which is equal to dynamically changing the effective sampling frequency seen in the baseband. Therefore, the opportunity now is to use the SRCs also for the timing adjustment in the time-continuous 3.5G signal. For this purpose, information about the timing deviation is calculated in the baseband IC based on parameters provided by the delay profile estimator (DPE) and forwarded via the digital RF interface into the RF transceiver. This information is used to dynamically modify the converting ratio of the SRCs for a defined period which leads to a virtually increased or decreased sampling frequency. By this means the time axis of a signal can be squeezed or stretched and therefore a time shift to advance or retard a time continuous signal can be applied. This effect can be used to adjust very smoothly the timing of 3.5G signals without loosing or adding any signal content in very fine time steps. By this means also a constant number of samples per frame is achieved for baseband processing. *Example:* Transmit time shift of a continuous 3.5G signal to a later time (Fig. 5) using a SRC in the transmit path (Fig. 6).

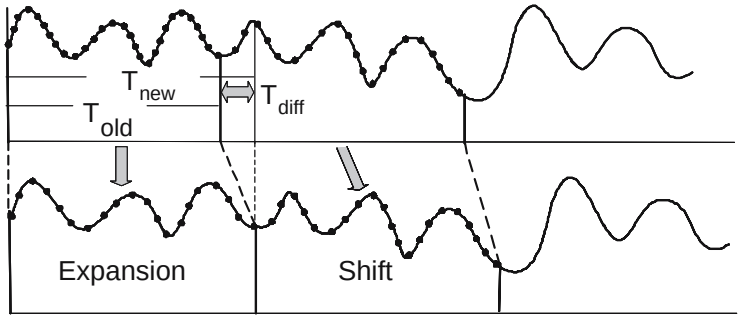


Fig. 5. Transmit time shift of a continuous 3.5G signal to a later time

- Reduction of sample rate at TX SRC input and constant sample rate at TX SRC output
- More output transmit samples have to be interpolated
- Signal content is expanded to a longer time period
- Then conversion ratio of SRC is switched back to standard ratio

Following the above steps a time shift ot the transmit signal (TX) to a later time without adding any content is achieved. By applying the follwing instructions a transmit time shift of the continuous 3.5G signal to an earlier time without deleting any content is carried out.

- Increase of sample rate at TX SRC input and constant sample rate at TX SRC output
- Less output transmit samples have to be interpolated
- Signal content is compressed to a shorter time period
- Then conversion ratio of SRC is switched back to standard ratio

In a similar manner time shifts can be implemented in the receive path (RX).

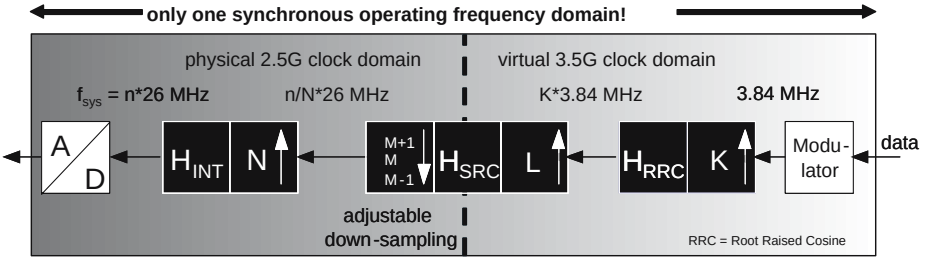


Fig. 6. Adaptation of virtual 3.5G symbol based sampling grid to physical grid based on 2.5G system specific PLL clock with SRC; with $n = 4 \rightarrow f_{sys} = 104$ MHz

5 SRC Architecture

Due to the high time resolution needed for the nearly asynchronous interpolation the impulse response of the SRC filter – based on an FIR structure – has many coefficients (Fig. 7). The number is given by the product of the numerator L of the conversion ratio $k = L/M = T_{A,in}/T_{A,out}$ times the number of considered input samples R. However, if the filter is decomposed into L polyphases only R ($R = 2 \ldots 8$) of them are needed for calculation of one output sample, i.e. one polyphase. Usually, only a small part of the SRC filter coefficients have to be stored if a simple trick is used: Due to the special low pass characteristics of the impulse response [2] it can be well approximated via linear segments and therefore it is sufficient to store only the coefficients at the end of these segments. The missing coefficients can be easily approximated by linear interpolation during runtime. The effective filter complexity is reduced to R linear coefficient interpolations and the convolution of one polyphase of the impulse response for every output sample, i.e. 6 filter taps in the given example.

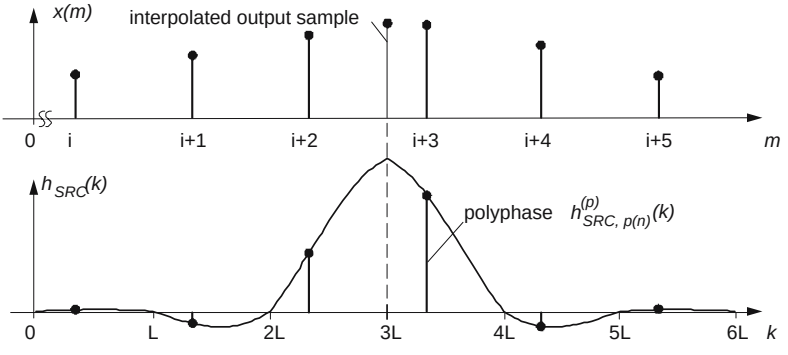


Fig. 7. Polyphase decomposition of interpolation filter; *Example*: $R = 6$ coefficients per polyphase

The transition between the nearly incommensurable input and output time grids m and n is shown in Fig. 8. L determines the resolution of the output sample position between two input samples. As also shown in Fig. 7 for each output sample R samples of the input signal are used. The indexes $m(n)$ of the first input sample of a group and the filter polyphase index $p(n)$ can be derived from the output sample index n by a polyphase calculator.

If we define Ω as the normalized sampling frequency at the output and Ω' at the input the magnitude response $Y(e^{j\Omega})$ of the SRC's output as a function of the input magnitude response $X(e^{j\Omega'})$ is given according [1] by

$$Y(e^{i\Omega}) = \sum_{\mu=-\infty}^{\infty} X(e^{j(\Omega-2\pi\mu)k})\, H_{\mathrm{SRC}}(e^{j(\Omega-2\pi\mu)k/L})\, si(\frac{1}{2}(\Omega - 2\pi\mu)k/L). \quad (1)$$

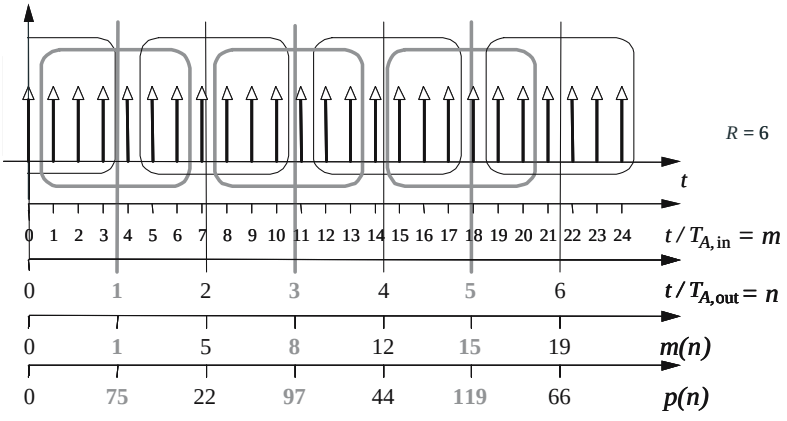


Fig. 8. Mapping incommensurable time grids; *Example*: $k = 128/459$

As depicted in Fig. 9 the SRC comprises a polyphase calculator containing an integrator – based on a modulo-adder – which is used to recursively calculate depending on the current ratio k the sampling phase of the next output sample $m(n+1)$ from the phase of previous output sample $m(n,k)$. This phase is expressed on the basis of the input time grid m. From this integrator value also the polyphase index $p(n+1,k)$ can be derived according $m(n,k) = (m(n-1)+M(k))$ div $L(k)$ and $p(n,k) = (p(n-1) + M(k))$ mod $L(k)$. p is used to select the polyphase of the SRC filter.

Furthermore, the SRC comprises an address generator to fetch the coeffients from the look-up table (LUT) and to control the linear interpolator. The linear interpolator calculates the coefficients of the polyphase based on the values provided by the LUT. The filter section is suited to calculate R filter taps per output sample. A finite state machine (FSM) is for control of the building blocks.

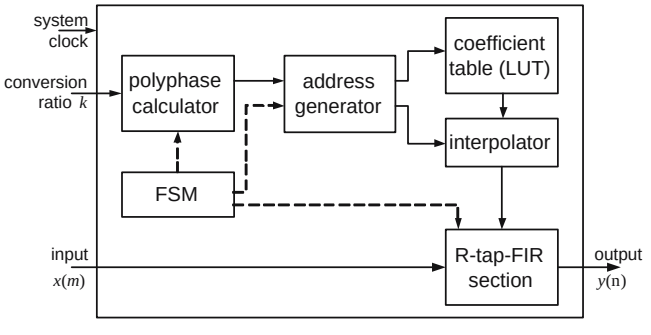


Fig. 9. Building blocks of the asynchronous sample rate converter

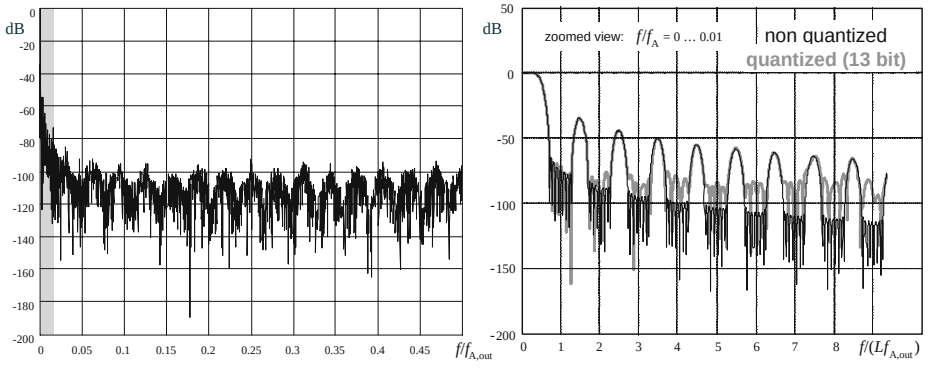


Fig. 10. SRC magnitude response has multiple stop bands

The quality of the SRC is visualized in Fig. 10. In the left part the overall magnitude response with multiple (non visible) stopbands is shown. Therefore, in the right part there is a zoomed view on the first stopbands from the L stopbands in total ($L = 128$ in this example). A very low passband ripple of less than 0.01 dB and a S/N for a full scale sinusoidal 8 bit output of 48.6 dB (THD+N) is achieved within the 3.5G bandwidth.

6 Conclusion

Two problems in multimode terminals are efficiently solved: covering 2.5G and 3.5G with one system clock *and* timing adjustment in time-continuous 3.5G signals. Furthermore, one fractional clock PLL is saved and reuse of the hardware for 2.5G and 3.5G signal processing is possible. Finally, the system clock frequency is not dependent on the wireless standards' symbol rates.

References

1. Wenzel, D., Speidel, J.: A digital Asynchronous Sample-Rate Converter for Digital Video Signals. IEEE Transactions on Consumer Electronics 46(1), 207–214 (2000)
2. Oetken, G., Parks, T.W., Schüßler, H.W.: New results in the design of digital interpolators. IEEE Transactions on Acoustics, Speech and Signal Processing 23, 301–309 (1975)
3. Rothacher, F.: Sample Rate Conversion: Algorithms and VLSI Implementation. Thesis, TH Zürich, Series in Microelectronics, vol. 44, Hartung-Gorre, Konstanz (1995)
4. Crochiere, R.E., Rabiner, L.R.: Multirate Digital Signal Processing. Prentice-Hall, Englewood Cliffs (1983)
5. Vaidyanathan, P.P.: Multirate Systems and Filter Banks. Prentice-Hall, Englewood Cliffs (1993)

Accelerating Space Variant Gaussian Filtering on Graphics Processing Unit

Roman Dudek, Carmelo Cuenca*, and Francisca Quintana**

Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria, Spain

Abstract. In this paper we examine the performance advantages of using a GPU to execute the space variant Gaussian filtering. Our results show that the straightforward convolution GPU implementation obtains up to 8 times better performance than the best recursive algorithm (the Deriche's filter) executed on a CPU, for useful maximum σ values. GPUs have turned out a useful option to obtain high execution performance, specially due to the emergence of high level languages for graphics hardware.

1 Introduction

In recent years the development of programmable graphics processors has placed the power of parallel computation in the hands of consumers. Graphics processing units (GPUs) have turned out useful coprocessors capable to perform more than the specific computations for which they were designed, and software developers are now paying attention to the general purpose computational ability of these GPUs and are using them in novel ways, specially due to the emergence of high level languages for graphics hardware [1,2].

Gaussian filtering is one of the most common effects in image processing. Due to the way the human vision system works it is frequently applied as a preprocessing stage in computer vision. As a result, the amount of image information is considerably reduced and the observer's attention is intentionally directed towards the important part of the scene.

Various algorithms were proposed and evaluated to efficiently perform Gaussian filtering [3]. However, not all these methods can be easily adapted for the filtering parameter varying from pixel to pixel. One method easily adapted is the straightforward usage of a convolution kernel for each pixel, using a different kernel for each pixel. However, this method is costly in terms of number of operations required, an important measure from the CPU execution time point of view. In order to reduce the number of operations, several methods have been proposed, possibly being most promising the recursive methods described in [3].

* Este trabajo ha sido financiado por la Consejería de Educación, Cultura y Deportes del Gobierno de Canarias mediante el proyecto PI2003/168-A.

** Este trabajo ha sido financiado por el Ministerio de Ciencia y Tecnología mediante el proyecto TIN-2004-07739-C02-01, por el HiPEAC European Network of Excellence y por el Barcelona Supercomputing Center.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 984–991, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Some of these methods can be adapted to work with a space variance, as shown in [3], where the adaptation of three different recursive filtering methods to use a varying blur radii (σ) are proposed and implemented. Among these recursive methods the Deriche's filter adapted for variable σ raises the best results on a CPU.

In this paper, we will examine the performance advantages of using a GPU to execute the straightforward convolution method, compared to the execution of the best recursive algorithm (the Deriche's filter) running on a CPU.

2 Space Variant Gaussian Filtering

The standard definition of an one dimensional discrete Gaussian convolution is given by

$$(f * g)(n) = \sum_m f(m) g_\sigma(n - m). \tag{1}$$

where the Gaussian kernel of scale σ is defined as

$$g_\sigma(n) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{n^2}{2\sigma^2}\right). \tag{2}$$

2.1 Deriche Filtering

In order to speed up Gaussian convolution several methods has been proposed and recursive approximations to Gaussian convolution has proved one of the most promising being Deriche's method one of the most popular. Deriche's method directly determines the recursive system

$$y_n = y_n^+ + y_n^-. \tag{3}$$

where

$$y_n^+ = n_0^+ x_n + n_1^+ x_{n-1} + n_2^+ x_{n-2} - d_1^+ y_{n-1}^+ - d_2^+ y_{n-2}^+ - d_3^+ y_{n-3}^+. \tag{4}$$

and

$$y_n^- = n_1^- x_{n+1} + n_2^- x_{n+2} + n_3^- x_{n+3} - d_1^- y_{n+1}^- - d_2^- y_{n+2}^- - d_3^- y_{n+3}^-. \tag{5}$$

where the coefficients are functions of the σ radii see [**?**].

3 Experiments

This section describes the experiments that have been carried out: We have implemented the Gaussian filtering on a CPU and on a GPU. We have tested both

implementations with a sequence of PAL (Phase Alternating Line) images, the standard TV definition mostly used in Europe. Each PAL image is represented by 3 color components (RGB), 8-bits-per-component, and a 720×576 pixel resolution.

The first experiment shows the application of the space variant Gaussian filtering on a long video sequence. The second one consists on measuring the comparative performance of the GPU and CPU implementations.

3.1 CPU Implementation

In the CPU implementation we ran the Deriche's recursive Gaussian filtering [3] adapted for an space variant σ on a 3.6 GHz Intel Xeon running Red Hat 9 Linux with kernel 2.4.26. The RAM memory size was 2 GB. The source code was compiled with the free GNU C Compiler.

Several optimizations were made to the source code in order to obtain the best CPU performance:

- **Streaming SIMD Extensions (SSE):** The Intel SSE extensions [5] were introduced in 1999. However, the GNU C compiler does nott take advantage of them in vector mode. In order to use the SSE extensions, inner loops were hand coded.
- **Data prefetching:** Data prefetching functions have been used in order to minimize cache miss latency by moving data into the processor cache before they are really accessed.
- **Precision versus Performance:** On one hand the recursive Gaussian convolution algorithm is precision sensitive. For any given arithmetics precision, there is a limit for the maximum value of σ that gives a correct result. On the other hand, the shorter the data size, the maximum use of the CPU's SIMD capabilities, and better performance results. As a compromise between precision and performance, and after running some tests, we have chosen SSE vectorized simple floating precision arithmetics, which warranties good SIMD usage and precise results for σ values up to approximately 80. In real applications these large σ values, or even larger, are seldom used. If needed, double precision arithmetics would be used, making the CPU implementation significantly slower.
- **Memory bandwidth:** Every image pixel needs its own σ which would define a matrix of floating point values per image. As 9 floating point coefficients per pixel are required, this array would be rather large, making memory bandwidth an important bottleneck. These coefficients can not be reasonably calculated per pixel from the σ value, as the equations to calculate them are complex (see [**?**]). To address this problem, the σ values were quantized to 256 possible states, ranging from $\sigma = 0$ to a maximum defined value. The quantization does not cause relevant precision reduction, and doing this two important optimizations were possible: Firstly, the σ values for each pixel can be stored in the otherwise unused alpha (A) channel of the RGBA image; and secondly, having only 256 possible different values of σ,

we can use a lookup of 256 different vector of the 9 parameters for Deriche's filter, instead of keeping full table of 9 values for every pixel of the image. Such lookup table does fit into the CPU's primary cache L1, reducing the used memory bandwidth.

3.2 GPU Implementation

Figure 1 is the NVIDIA GeForce 7800 GTX architecture [6] at a glance. At the top there are 8 separate vertex shader pipelines, leading to a setup and then a shader instruction dispatch. The dispatch is the logic that properly load balances the pixel shaders below it and optimize any shader instructions for optimal performance. Immediately following are the 24 pixel shaders that are organized into 6×4 partitions. Finally, the fragment crossbar routes the data into the 16 raster operators (ROPs) where it is combined with the data previously stored in the video memory, and written into the video memory over the 256-bits wide and broken into 4 64-bit partitions. The vertex shader units can handle 34.4 billion floating point operations per second and the pixel shaders can handle 278.6 billion operations per second.

The GPU's programming model is quite different to the CPU's programming model [7]. A GPU executes one kernel program (or fragment program) on a input data stream (or fragments). The fragment program is executed simultaneously on several pipes. Each pipe can typically execute up to two *madd* (multiply and add) operations per clock cycle. A typical general purpose GPU application loads data to be processed into a texture memory, one element per texture element (texel). This texture is then used as input while drawing a full-viewport quad, with dimensions same as that of the texture. This causes the fragment program to be executed for every resulting pixel. Arbitrary input texels can be used to calculate each output pixel, however 2-dimensional locality of access improves the performance due to better cache use.

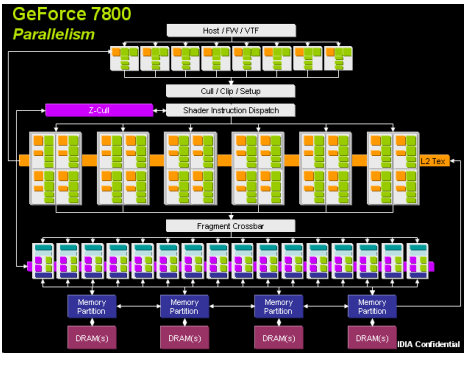


Fig. 1. NVIDIA GeForce 7800 GTX architecture

Programming these complex graphics hardware is becoming an easy task due to the emergence of high level languages for graphics hardware, like OpenGL Shading Language (GLSL) [2], NVIDIA's C for graphics (Cg) [8,?], and the Microsoft's High Level Shading Language (HLSL) [10]. Vertex and fragment programs are written in a C-like language, removing the need for any graphics-specific knowledge of GPU development.

We have used the Cg language to implement the GPU algorithm based on a straightforward convolution, performed per lines and per columns. The convolution was performed in 32-bit floating point precision, as in the CPU implementation. The potentially infinite convolution kernels were clamped at 3σ. We precalculate 256 different kernels for σ varying 0 to a maximum user-defined σ. We have made different measures for maximum σ varying from 1 to 80.

The GPU implementation was tested on the same 3.6 GHz Intel Xeon running Red Hat 9 Linux with kernel 2.4.26, as in the CPU implementation. We have tried two different graphics cards from NVIDIA: a GeForce 7800 GTX-512 PCIe and a GeForce 6600 GT PCIe, both of them with a display driver version 1.0-8178. Of course the first card is better than the second, but it is also more expensive. Moreover, the second card is a typical graphics card that can be found nowadays on every personal computer with some gaming capabilities. The GeForce 6600 GT results will point out how much performance improvement can a normal user expect from using his graphics hardware for general programming, while the GeForce 7800 GTX-512 results while show that it is worthwhile investing some extra money in a better graphics card if we intend to use it for general purpose programming.

4 Results

This section presents the experimental results. Firstly we will show the results that illustrate a practical application of Gaussian filtering. We took an image sequence from real life. Figure 2 shows a sample image from the sequence where a carnival scene with dancers appears. The experiment consisted in following one of the dancers along the sequence. This dancer is kept focused while her surroundings are gradually blurred in order to keep the observer's attention in the dancer. Figure 3 shows the same sample image as in Figure 2 after the space variant Gaussian filtering effect has been carried out. Finally, Figure 4 shows the foveation mask, that is, a black and white image that defines how much every pixel in Figure 2 should be blurred.

Secondly, we will present the performance measurements of the CPU and GPU implementations.

Figure 5 shows the typical foveation mask used to perform the execution time measurements, extracted from [3]. It is defined by the equation $f(r) = \frac{1-\cos\left(\frac{r\pi}{d}\right)}{2}$, where r is the distance from the pixel to the center of the mask, and d is the distance from the image corner to the center of the mask.

Figure 6 shows the results obtained when measuring the execution times of the space variant Gaussian filtering algorithms executed in the CPU and GPU



Fig. 2. Sample source image from a video sequence



Fig. 3. Sample result image after performing Gaussian filtering. The maximum σ was 7.2.

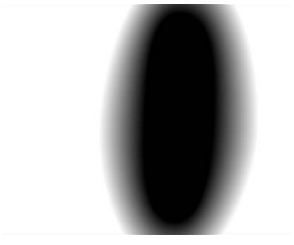


Fig. 4. The foveation mask: A black and white image defining how much every pixel has to be blurred

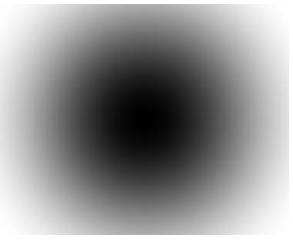


Fig. 5. Foveation mask used to perform execution time measurements

implementations as a function of the maximum σ used. There are three different plots: the 3.6 GHz Intel Xeon CPU recursive implementation, the NVIDIA 6600 GT GPU implementation and the NVIDIA 7800 GTX-512 GPU implementation. Several considerations can be pointed out when looking at this Figure. First of all, the execution time in the recursive CPU implementation does not depend on the σ value, as it was expected, and remains the same for any σ value. However, in both GPU implementations the execution time is proportional to σ, and it increases almost linearly with the maximum σ.

It is important to note that the CPU execution times can be considered meaningful only up to approximately $\sigma = 80$. Over this radius, result images contain large errors due to the 32-bit floating point arithmetic used. For larger σ values 64-bit floating point arithmetic would be required, thus increasing the execution times in the CPU implementation. However, in practice such large blurring is seldom needed. The convolution based GPU implementation is much less precision sensitive, with 32-bit arithmetic's sufficient for any senseful σ (More than 20000.0 by basic estimation).

We can also see that the GPU execution times are much lower than the CPU execution times. However for a certain σ the GPU execution time becomes higher than the CPU execution time. In the case of the NVIDIA GeForce 6600 GT GPU execution this happens at $\sigma = 20$, and in the NVIDIA GeForce 7800 GTX-512 it

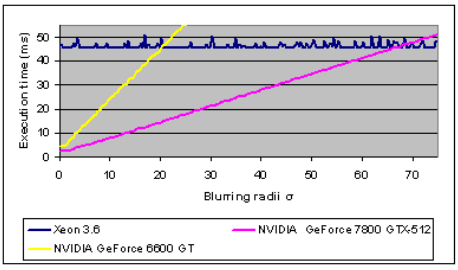


Fig. 6. Measured execution times (ms) in the CPU and GPU implementations as a function of σ

happens at $\sigma = 66$. However, practical σ values are not that high. As an example, in the carnival dancer experiment previously shown, a σ equals to 7.2 was used. For that σ value the NVIDIA GeForce 7800 GTX-512 GPU performs 7.5 times better than the CPU, and the NVIDIA GeForce 6600 GT performs 2.5 times better. For that reason we claim that the GPU implementation outperforms the CPU recursive implementation in useful cases, even with an inexpensive low-end GPU.

It is also important to note that the GPU performs quicker than the CPU, even while computing a much higher number of operations than the straightforward convolution algorithm implemented in the GPU. We can estimate the number of floating point operations in each implementation by assuming a fixed σ value, for example $\sigma = 16$, constant for each pixel of the image. In the GPU implementation, for $\sigma = 16$, cut at 3σ, which means 95-pixels kernel size. Three add operations and three multiply operations are carried out for each kernel element, and the kernel is applied twice (horizontal + vertical). The image has 720×576 pixels, thus making a total of $95 \times 6 \times 2 \times 720 \times 576 = 472$ million floating point operations (MFLOPs) performed. In the CPU implementation 23 floating point operations per pixel and component are needed, and there are three components in each pixel. The iteration is applied twice (horizontal + vertical). The image has 720×576 pixels, thus making a total of $23 \times 3 \times 2 \times 720 \times 576 = 57.2$ MFLOPs performed. Therefore, the GPU implementation carries out more than 8 times more floating point operations than the CPU implementation and it is still able to obtain a substantially lower execution time. While the CPU implementation needs to perform much less operations thanks to its more efficient algorithm, the GPU implementation still performs better due its raw processing power and its good mapping of image processing applications into its architecture.

Finally, we want to mention that in the GPU implementation, measured times assume that input images are already in the graphics memory, and results are also stored in the graphics memory. If they were in the system memory, then we must add the transfer time needed to upload the image and the convolution table to the graphics memory and to download the result image back to the system memory. Our measurements for these extra times are under 4 ms. However, data transfer is not always needed in a practical application, as the filter can

be concatenated with other GPU based processes, or results are intended to be displayed on the screen.

5 Conclusions

Results show that for useful σ values the GPU implementations outperforms the CPU implementation, even in the case of an inexpensive GPU. At a certain point, the CPU would eventually overtake GPU as the fastest performing algorithm. However, this overtake happens for impractical large values of σ. For usual values of σ, for both scientific and artistic uses, the GPU performs considerably better.

The GPU processing time rises approximately linearly with the maximum σ, as larger kernels are needed. However, for practical σ values, we measured 2.5x-7.5x better performance on GPU compared to a CPU implementation, even if the data transfer times penalty is applied.

All in all, GPUs are an inexpensive option to obtain high performance execution times with little specific knowledge on hardware graphics. Moreover, in recent years GPUs are improving at a much higher rate than CPUs. For that reason it is likely that a GPU design is a suitable engine to process images in a lower time.

References

1. Fernando, R., Kilgard, M.: The Cg Tutorial: The Definitive Guide to Programmable Realtime Graphics. Addison-Wesley, London, UK (2003)
2. Kessenich, J., Baldwin, D., Rost, R.: The OpenGL® Shading Language, available: `http://oss.sgi.com/projects/ogl-sample/registry/ARB/GLSLangSpec.Full.1.10.59.pdf, 2004`
3. Tan, S., Dale, J.L., Johnston, A.: Performance of three recursive algorithms for fast space-variant Gaussian filtering. Real-Time Imaging 9(3), 215–228 (2003)
4. Buck, I., et al.: Brook for GPUs: Stream Computing on Graphics Hardware. ACM Transactions on Graphics (August 2004)
5. Gerber, R.: Software Optimization Cookbook: High-Performance Recipes for the Intel® Architecture Intel Press 2005.
6. (2005), `http://www.nvidia.com/page/geforce_7800.html`
7. Buck, I., Purcell, T.: A Toolkit for Computation on GPUs. In: Fernado, R. (ed.) GPU Gems. Programming Techniques, Tips, and Tricks for Real-Time Graphics, pp. 621–636. Addison-Wesley, Boston (2004)
8. Fernando, R., Kilgard, M.J.: The Cg Tutorial. The Definitive Guide to Programmable Real-Time Graphics. Addison-Wesley Press, London, UK (2003)
9. Mark, W., Glanville, S., Akeley, K.: Cg: A system for programming graphics hardware in a C-like language. ACM Transactions on Graphics (August 2003)
10. St-Laurent, S.: The COMPLETE Effect and HLSL Guide Paradoxal Press (2005)

Ant-Based Topology Convergence Algorithms for Resource Management in VANETs

Frank Chiang, Zenon Chaczko, Johnson Agbinya, and Robin Braun

Faculty of Engineering, University of Technology Sydney,
Broadway, NSW 2007, Australia
frankj@eng.uts.edu.au

Abstract. Frequent changes caused by IP-connectivity and user-oriented services in Inter-Vehicular Communication Networks (VCNs) set great challenges to construct reliable, secure and fast converged topology formed by trusted mobile nodes and links. In this paper, based on a new metric for network performance called topology convergence and a new Object-Oriented Management Information Base - active MIB (O:MIB), we propose an ant-based topology convergence algorithm that applies the swarm intelligence metaphor to find the near-optimal converged topology in VCNs which maximizes system performance and guarantee a further sustainable and maintainable system topology to achieve Quality of Service (QoS) and system throughput. This algorithm is essentially a distributed approach in that each node collects information from local neighbor nodes by invoking the methods from each localized O:MIB, through the sending and receiving of ant packets from each active node, to find the appropriate nodes to construct a routing path. Simulation results show this approach can lead to a fast converged topology with regards to multiple optimization objectives, as well as scale to network sizes and service demands.

1 Introduction

Vehicular Ad-hoc Communication Networks (VANETs) are becoming an active area in telecommunication research community [1]. As a Wireless LAN incorporating Mobile Ad-hoc Networks (MANETs), VCNs target the design and deployment of the QoS-assured, secure and high-speed communication connectivities on mobile platforms such as buses, trains, cars, ships, marine squad of battlefields, where dispersed portable devices are enabled to establish on-demand pervasive communications in decentralized manners. Inter-vehicular, Intra-vehicular ad-hoc mobile networks and vehicular-to-Internet communication networks are indispensable parts for the envisaged ubiquitous communication scenario, this adds more alternative ways for internetworking of WLANs, Satellite and Cellular systems. Currently, vehicular communications are typically required to: (1) construct and maintain QoS-assured links, treat dynamic/static workload fairly and further assign reasonable resources to support user-oriented services; (2) provide acceptable reliability; (3) secure communication links and provide fault-tolerance and self-restoration mechanism; (4) scale to future expected network expansions and growths.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 992–1000, 2007.
© Springer-Verlag Berlin Heidelberg 2007

To incorporate all these issues, topology convergence aims to find the converged topology which maximizes system capacity and throughput by minimizing interference and management costs, optimizing node cooperation and efficiently utilizing battery power as much as possible to self-adapting the topology such that the required QoS and security concerns are satisfied.

1.1 Topology Control vs Topology Convergence

A huge body of literature has been dedicated to topology control. However, topology convergence is different from the topology control. Topology control targets the maintenance of the chosen topology in a wireless network by adjusting transmission power at each node [2] and such that an objective function of the transmission powers are optimized. While topology convergence targets the convergence of mobile hotspots or Access points internetworking paths to an optimal topology not only before but also while the maintenance phase is invoked. By establishing/updating a fast, reliable, secure and cost-effective topology under multiple QoS constraints and management costs, it must be able to autonomically survive any hazards, resist any malicious attack, show certain adaptability when roaming from one wireless LAN to another type of wireless LAN. And system throughput, data routing and security requirements are achieved afterwards. Several successive stages leading to topology convergence include (1) distributed topology discovery (e.g., gateways discovery for mobile hotspots); (2) multiple objectives optimization process; (3) behavior/pattern matching with a known optimal topology; (4) configuration/reconfiguation/activation. However the nature of the formation of a converged topology in complex, dynamic, heterogeneous environments is unknown yet. This paper seems to be a first attempt in exploring this domain and carry out a preliminary simulation test to better understand the contributions from node cooperations as well as to understand the effects of the convergence on network performance.

1.2 Research Question and Contributions

It is the authors' belief that the formation of the converged topology should relate to the (1) routing mechanism (2) Availability of IP-connectivity (3) Optimal on-demand service-instantiated topology (4) Security issues. Our main research question is: *how do we address these new Network Management challenges, factoring in the ingredients of a distributed environment, to construct efficient, robust, secure, scalable and mobile VANETs that are capable of providing the ubiquitous connectivity required by multimedia applications such as VoIP and TVoIP? We argue that autonomic management paradigm can be the solution to scalability and QoS requirements for distributed VANETs.*

Specifically, our goals are to address the unexplored nature of topology convergence for VANETs by creating a new routing protocol - an ant-based topology convergence algorithm based on resource orientated cost function, topped with an efficient self-managed network management paradigm. This new routing protocol is unlike conventional shortest path-based routing algorithms such as DSR

[3]. The multimedia streaming experimental results test performance of the QoS of routing protocols for VANETs. We believe that our 4/3 management structure [4], combined with the notions of an proposed O:MIB and biomimetic learning and adaptation strategies will help to resolve the topology convergence issue.

The remainder of the paper is organized as follows. Section 2 presents the ant colony biological model. Section 3 describes our algorithm and its application to learning and adaptation. As a validation test, the performance comparison in Section 4 shows the effectiveness and robustness to guide QoS-assured topology convergence process in VCNs with respects to the known testing metrics. Conclusion and future development are stated in section 5.

2 Ant Colony Biological Model

2.1 Ant Agent Behavior Model Mapping to Network OSI Model

The *ant agent behavioral model* is inspired from the foraging behavior of an ant colony. The ant model is working between the *networking* layer and the *application* layer in order to improve the routing performance of IP data packet in VCNs. In doing so, this biologically-inspired scheme can be seamlessly integrated into current Internet infrastructure economically and flexibly.

A fully devised mobile vehicle having wireless bi-directional connectivity to Base Station/Satellite ca classified as a node. Each node in VCNs contains one ant colony, which can be classified into 4 parts: (1) packets queue repository, (2) application/service repository, (3) communication channels and (4) Heterarchy ant agents in the container.

The structure of an ant nest is shown in Figure 1. The packet queues repository is the interface to the PHY and MAC layer while application/service queue repository is the interface to the application layer. All the packets are sent out or received in through the ant nest interfaces. The information flows go through the environment inside and outside ant colony all the time, which facilitates ant agents to communicate with each other through *communication channels*. The communication channels include two modes: (1) indirect stigmergic communication channel and (2) direct contact communications. The ant agent container contains *heterarchy* ant colony, where ant colony is constructed via a *heterarchical* way but not in a *hierarchical* way as described by Wilson & Holldobler in [5] and Dréo in [6]. Heterarchical structure makes the ants communication with each other close without the level restriction as of hierarchical network. Heterarchy functions as a bottom-up designed system, where the system is highly self-organized without the centralized control, emergent properties are emerged in that each entity operates by following simple rules, when the whole colony population is self-organized to work together and finally exhibits desirable patterns and behaviors to reach system goals. While *hierarchical* structure follows a top-down design, communications are restricted to go through various levels

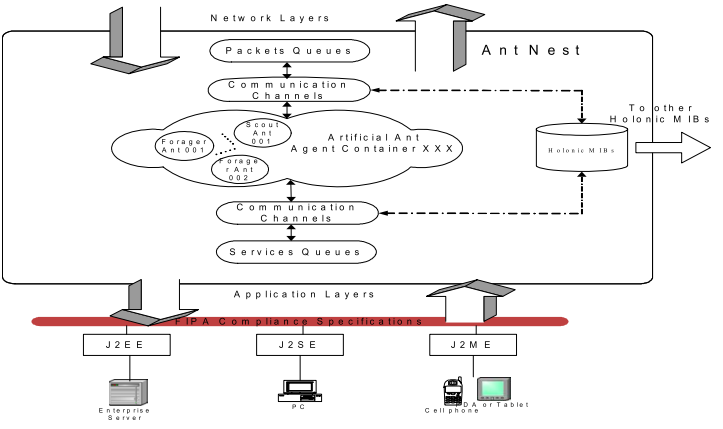


Fig. 1. Ant Nest Model

to reach the final destination, and most of the time, a centralized control may be needed.

2.2 Conventional MIB vs Proposed Active O:MIB

Inspired from Active Networks [7], we propose an active O:MIB which is different than traditional MIB in terms of the structure and its functionality. Object-Oriented paradigms are applied into the construction of conventional MIBs who possess holonic hierachy[1].

3 Constrained Ant-Based TC Protocols

The *off-line* ACO optimization applied into service management domain has been tested in our paper [10] . This paper attempts to try *online* optimization instead in the Network layer. The information interchanging are done in real-time mode by the localized autonomic entity - O:MIBs which reside on each network devices and services. Each local O:MIB establishes an information environment where an artificial ant colony lives. The network system consisting of many local O:MIB comes into being an *information ecosystem*, which mimicks the behaviors and patterns of the nature ecosystem. Network activities such as IP packet transmission, forwarding, management information collection, decision making processes are all happening in this ecosystem. Recent work [11] shows that the congestion in manhattan networks are often undesirably worse in that the large number of vehicular access points aggregated into some areas such as the crossing area. The performance of getting fast converged topology for communications can be improved by the proposed ant-based algorithms, which works according to the network status, traffic balancing behaviors can be expected.

[1] Due to the page limits, full details of O:MIB are available in our papers [8] [9].

3.1 Ant-Based TC Protocol

As we design in previous section, each vehicular node i could be either source node or destination node and functions asynchronously and independently from each other, the node can interact with other nodes via sending and receiving ant packets which is similar to the mechanism used in active networks. This protocol is driven by on-demand events which can be roughly classified into four stages: (1) Initialized by user-oriented on-demand service requests (2) Checking local active O:MIB. (2) Preconfigured periods of ant packets sending and receiving. (3) Update pheromone trails (4) Start another round of searching cycle in case of unexpected disruptive events. Each node i needs to maintain a pheromone table which contains time-varying pheromone value $\tau(C_{E_{ij}}, t)$ for different $C_{E_{ij}}(\cdot)$ level. We use three different levels $\omega_1(C_{E_{ij}}(\cdot))$, $\omega_2(C_{E_{ij}}(\cdot)), \omega_3(C_{E_{ij}}(\cdot))$ to simply distinguish them as the weights for pheromone calculations and set the rules accordingly for pheromone updating process. Table 1 shows the pheromone trails for *node* 1 with its all neighbor nodes.

Table 1. Pheromone Table

Node n_1	Neighbour n_2	Neighbour n_3	Neighbour n_4	Neighbour n_5	Neighbour n_6
$C^{\omega_1}_{E_{ij}}$	$\tau_{n_1,n_2}(C^{\omega_1}_{E_{1,2}}, t)$	$\tau_{n_1,n_3}(C^{\omega_1}_{E_{1,3}}, t)$	$\tau_{n_1,n_4}(C^{\omega_1}_{E_{1,4}}, t)$	$\tau_{n_1,n_5}(C^{\omega_1}_{E_{1,5}}, t)$	$\tau_{n_1,n_6}(C^{\omega_1}_{E_{1,6}}, t)$
$C^{\omega_2}_{E_{ij}}$	$\tau_{n_1,n_2}(C^{\omega_2}_{E_{1,2}}, t)$	$\tau_{n_1,n_3}(C^{\omega_2}_{E_{1,3}}, t)$	$\tau_{n_1,n_4}(C^{\omega_2}_{E_{1,4}}, t)$	$\tau_{n_1,n_5}(C^{\omega_2}_{E_{1,5}}, t)$	$\tau_{n_1,n_6}(C^{\omega_2}_{E_{1,6}}, t)$
$C^{\omega_3}_{E_{ij}}$	$\tau_{n_1,n_2}(C^{\omega_3}_{E_{1,2}}, t)$	$\tau_{n_1,n_3}(C^{\omega_3}_{E_{1,3}}, t)$	$\tau_{n_1,n_4}(C^{\omega_3}_{E_{1,4}}, t)$	$\tau_{n_1,n_5}(C^{\omega_3}_{E_{1,5}}, t)$	$\tau_{n_1,n_6}(C^{\omega_3}_{E_{1,6}}, t)$

The edge parameter $C_{E(i,j)}$ between vehicles denote a function of multiple factors that the topology convergence process needs to cover. The optimal convergence process is then transformed into an ant traversing through the graphical network where a certain number of vehicle nodes are visited until the stopping criteria are reached. Moreover, each node is assumed to be traversed once at most.

Suppose the ant is randomly placed on node i, the probability of an ant k choosing the next adjacent node j as denoted in [12] is:

$$P_k(i,j) = \begin{cases} \dfrac{[\tau_{i,j}]^{\alpha}\left[\eta_{i,j}\right]^{\beta}}{\sum_{l \in U}[\tau_{i,l}]^{\alpha}\left[\eta_{i,l}\right]^{\beta}} & \text{if } j \in U \text{ and } l \neq j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $\eta_{i,j}$ is called *visibility* and represents the *heuristic desirability; U* is the set of neighbor nodes of node i that has not transpassed, and U is the set of feasible components which can be potentially selected to fulfill a successful configuration process. In our present method, information from holonic MIBs contribute to the main part of $\eta_{i,j}$, denoted as $\eta_{i,j} \triangleq \frac{1}{C_{E(i,j)}}$. The main steps of our proposed method for the topological convergence process are summarized as follows:

Step 1) Initialize cost values and ACO parameters, such as predefined iteration times t, initial pheromone values $\tau_{i,j}(t)$ and stagnation conditions, etc.

(1) Create M artificial ant agents for one ant colony, where M is usually set to be equal to the number of nodes; Each ant colony is used to instantiate one topology. (2) Set initial iteration variable $t_0 = 0$; set predefined iteration times t=200; and set the initial pheromone value in all trails equal to a constant initial value $\tau_{i,j}(0) = C$. C should not be zero because a non-zero finite constant could delay the process of evaporation into zero pheromone density so as to further avoid the earlier local optima. (3) Dispatch each ant in this ant colony from one node. The ant retrieves *heuristic* information from Holonic MIB, e.g., available bandwidth information, delay information, capacity and cost information, connectivity information.

Step 2) Build the Solution

M ants start to follow the transition rule to build solutions asynchronously. Meanwhile, heuristic information, as a transition constraint, ensures ants should only traverse the *feasible* nodes in our problem domain before making a decision of a move. Whenever the period of predefined time is due or the receiving of the ant packets which require certain actions to be taken.

The edge parameter $C_{E(i,j)}(t)$ is calculated and usually less than 1. Until now, all the *heuristic* information and edge vector values are determinate. The probability of selecting next relaying vehicle at one iteration step is recorded into a table. When the destination nodes are reached by all M ants, we consider it as the completion of one round of iterations. There are 3 possibly successful ant traversing scenarios: 1) Ant traverses through possible nodes until it reaches destination node i and sleep there. 2) Ant reaches the destination nodes within the maximum limits of hops counts **H**. 3) Suppose some node goes faulty, the ants will be awaked up on this node to find an *alternative* node from the same class of nodes as the faulty nodes, or the next available ant will find an alternative node.

Let Pr($n-$th Ant) represent the probability of being selected by the n-th ant; the n-th ant in node i will select the next node j to visit with probability equation (1)

Not all ants are able to finish a complete traverse path from product to resources. For ants which are able to finish traverse paths during this iteration, we *match* the components with the nodes on paths and *record* the minimum sum value for $C_{E(i,j)}(t)$

Step 3) Update pheromone trails

Upon receiving the ant packet from node i, equation (2) is applied to increase pheromone value along the path. The insertions and reductions will influence the searching process for the next iteration. We adopt the conventional ACO pheromone update rule as:

– Intensify pheromone: an absolute amount of pheromone $\triangle\tau^k_{i,j} > 0$ is added to the existing pheromone values on the paths completed by k-th ant

$$\triangle\tau^k_{i,j} = \begin{cases} \frac{Q}{Z_{\min(k)}} & if\ (i,j) \in T_k \\ 0 & if \notin T_k \end{cases} \tag{2}$$

Table 2. Speed-Density in Syndey CBD

VD(Vehicles/Km)	Speed(Km/h)
40	58
80	52
120	40
240	10

where T_k are the paths having been traversed by k-th ant. Q is a constant, more research on Q value have been done in Dorigo [13].

- Evaporate pheromone: When the wireless *link life time* is due or no more packets go through, the evapration occurs as stated in equation 3.

$$\tau_{i,j} = (1 - \rho) \times \tau_{i,j} \qquad \forall i, j \in [1 : m] \tag{3}$$

where $\rho \in [0,\ 1)$ is the pheromone evaporation rate. $\tau_{i,j}$is the current pheromone density on the trail $a(i, j)$.

- Mutate pheromone update process

Step 4) Start another round of iterations

Before the predefined searching iteration times are reached, another round of iterations always starts again, immediately after the previous round of iterations ends. Meanwhile, the predefined iteration times t is increased by 1: $t = t+1$. Go to step 2).

4 Simulation

4.1 Simulation Platform and Configuration Parameters

Vehicle Density (VD) in a lane is defined as the number of vehicles in a single lane (n/km), hence, the maximum VD in a single lane can be calculated as: $Max\{VD\}_{lane} = \frac{L}{l}$

The recent extension work by Treiber and Helbing [14], who developed the highway car following model further into a community *intelligent driver model* (IDM) is given as:$\frac{dv_j}{dt} = a_{\max}\left[1 - \left(\frac{v_j}{v_{desired}}\right)^{\lambda} - \left(\frac{d_{desired}(v_j, \triangle v_j)}{d_j}\right)^2\right]$.

We implement the community mobility model into a simulation scenario that emulates the real traffic patterns in community road with various driving states. Based on this simulator, we apply our proposed scalable ant-based TC protocols into this dynamic road condition and compare the performance of topology convergence with other existing routing protocols such as DSR, AODV.

Metrics used in evaluating the performance of topology convergence for proposed ant-based TC protocols: 1.) Packet Delivery Ratio; 2.) Message Overheads; 3.) Delay; 4.) System throughput and capacity.

4.2 Experiment Results

As for the conventional DSR and AODV, we consider the 3 performance measures: (1) Packet Delivery Ratio, (2) Message Overheads, (3) end-to-end Delays. They are shown in the following figures. We compare the proposed TC algorithm based on those metrics and prove it can efficiently improve the packet delivery ratio and system throughput/capacity, and robustness and scalability can be achieved too. Figure 2 and Figure 3 presents the simulation comparison results on Packet Delivery Ratio (PDR) and end-to-end average delay between our proposed scheme and AODV routing scheme.

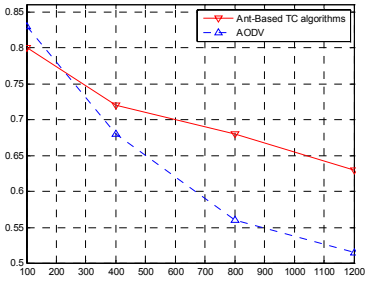


Fig. 2. Comparison of PDR

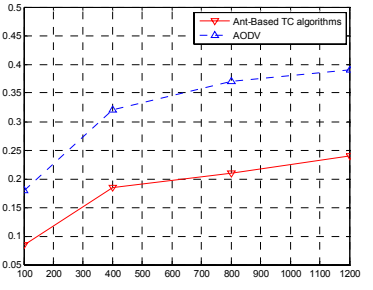


Fig. 3. Comparison of Average Packet Delay

5 Conclusion and Future Work

This paper proposed ant-based topology convergence algorithm to guide QoS-assured vehicular communication in future complex wireless networks. An innovative active O:MIB is designed and implemented. The efficiency of the approach outperform the current DSR and AODV. Future work includes: (1)Bi-directional vehicular communications to be implemented into our simulator; (2) Scalability and security issues to be taken into consideration.

References

1. Chisalita, I., Shahmehri, N.: A novel architecture for supporting vehicular communication. In: Proceedings of IEEE 56th Vehicular Technology Conference, vol. 2, pp. 1002–1006 (2002)
2. Ramanathan, R., Rosales-Hain, R.: Topology control of multihop wireless networks using transmit power adjustment. In: Proceedings of IEEE Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 2, pp. 404–413 (2000)
3. Johnson, D.B., Maltz, D.A.: Dynamic source routing in ad hoc wireless networks. Mobile Computing 5, 153–181 (1996)

4. Chiang, F., Braun, R.: Towards a management paradigm with a constrained benchmark for autonomic communications. In: Cheung, Y.-m. (ed.) Proceedings of the 2006 International Conference on Computational Intelligence and Security, vol. I, pp. 520–523 (2006)
5. Wilson, E.O., Holldobler, B.: Dense heterarchy and mass communication as the basis of organization in ant colonies. Trend in Ecology and Evolution 3, 65–68 (1988)
6. Dreo, J., Siarry, P.: A new ant colony algorithm using the heterarchical concept aimed at optimization of multiminima continuous functions. In: Dorigo, M., Di Caro, G.A., Sampels, M. (eds.) Ant Algorithms. LNCS, vol. 2463, Springer, Heidelberg (2002)
7. Bush, S.F., Kulkarni, A.: Active Networks and Active Network Management: A Proactive Management Framework. Kluwer Academic/Plenum Publishers (2001)
8. Chiang, F.: Risk and vulnerability assessment of secure autonomic communication networks. In: Wireless 2007. Proceedings of the 2nd International Conference on Wireless Broadband and Ultra Wideband Communication (2007)
9. Chaczko, Z., Chiang, F., Braun, R.: Active mib: Addressing challenges of wireless mesh networks. Lecture Notes in Computer Sciences (2007)
10. Chiang, F., et al.: Self-configuration of network services with nature-inspired learning and adaptation. Journal of Network and Systems Management 15, 87–116 (2007)
11. Guzman, G.N., Agbinya, J.: Topology analysis of moving wireless networks. In: Auswireless 2006 (2006)
12. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Santa Fe Institute Studies in the Sciences of Complexity. Oxford University Press, Oxford (1999)
13. Dorigo, M., Stuzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
14. Treiber, M., Hennecke, A., Helbing, D.: Congested traffic states in empirical observations and microscopic simulations. Physical Review E 62 1805 (2000)

Simulation of a Signal Arbitration Algorithm for a Sensor Array

Octavian Mocanu and Joan Oliver

ETSE – UAB, Dept. of Microelectronics and Electronic Systems Engineering
Edifici Q, 08193 - Bellaterra (Barcelona), Spain
{Octavian.Mocanu, Joan.Oliver}@uab.es

Abstract. We present a simulation based study of an arrangement consisting of a sensor array, connected to a set of signal processing units (PUs), capable to be reconfigured in real time, depending on the signal type and processing requirements. The signal issued by a sensor is conveyed by means of an arbitration algorithm to a proper PU, so that the number of its state transitions should decrease. The algorithm also ensures a high throughput of the signals to be processed. The system includes queues at the PU level, to lower data loss. The functioning of the PUs is based upon two strategies: maximising task charge and standby state period, to maintain the highest number of spare PUs in order to decrease the reconfiguration actions and the associate power penalty. The sensors are capable to issue signals with variable frequency in time, a factor considered when prioritising among concomitantly generated signals.

Keywords: Arbitration, Processor unit, Reconfiguration, Sensor array.

1 Introduction

Applications involving sensors [1, 2] are already an everyday reality, and with respect to this, there are some aspects of which the energy consumption belongs to the most critical ones. The energy reduction techniques are manifold, being in a simpler manner distinguished between the hardware and the software level [3].

The latter one is built by at least three layers: the operating system, the programming languages being used and the compilers, and it is not going to be addressed here. The former level includes the physical layer, where electrical and mechanical properties are envisaged, and the data link and network layers, where beside topics as framing and contention, the interconnection topology, packet addressing or multiple channel environment are study objects [4-8].

On the other hand, the possibility that a processing unit (PU), which performs operations optimally tailored from the perspective of the energy consumption (e.g. [9]), be reconfigured for various tasks, depending on a particular need, is a topic of high interest [10, 11]. The subject of how to convey the data to the PUs, using queuing strategies in order to minimise the energy consumed, but nevertheless keep the real time advantage is of great importance in such a system [12, 13].

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1001 – 1008, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In our application, we assume to have a heterogeneous sensor set, meaning that the type of the generated signals and their issuing frequency can be variable. For the moment, we consider them digital. The signals are conveyed to a processing unit array, under the supervision of an arbitration circuit, in order to be evaluated and processed.

To reduce the energy consumption, we design in such a way the arbitration circuit, that the allocation of each task (signal) to a PU be overall optimum, in the sense that a PU should change its (processing type) state as few times as possible, unless it gets the standby state. (It is assumed that, if needed, a PU can be reconfigured, to perform a new computation task from a given set.) Therefore, we develop an algorithm that searches for an adequate PU for an incoming signal, at the same time, trying to avoid any data loss.

A discussion about how to connect the PUs, viewed as a future goal of our work, ends up the presentation.

2 General Considerations

Prior to its transmission by a sensor, a signal is packed to form a frame consisting of a token and the actual data value (Fig. 1).

The token identifies the type of the signal, as well as the signal issuing frequency. The latter may represent a pre-computed estimate or the runtime obtained value, assuming variable behaviour in time.

The data packet is then conveyed towards the PU. The destination it is assigned to depends on its type and the availability of a PU. The decision is taken by the arbitration unit, which follows the algorithm exposed in section 3.

Every PU has associated a queue. Moreover, one can append a supplementary queue to every sensor, prior to the arbitration process, unless the data may be dropped. If there are two or more data packets that arrive simultaneously and should be assigned to a PU of the same type, the signal frequency field of the signal token fixes the processing priority to the data with higher generation frequency.



Fig. 1. Data Packet Structure

3 Arbitration Algorithm

3.1 Algorithm Description

The main idea of the arbitration algorithm is: 1) to preferably use a same PU, assigned to a given task type, for as many data requests as possible, in order to

minimise the number of state transitions, and thus the energy consumption, and 2) to leave a PU as long as possible in the steady state, with the same goal in view. On the other hand, the reconfigurability of the PU offers an increased flexibility and velocity of computation.

There are three structures implied in the arbitration scheme: the sensor set, the processing unit array (or core array), and the queues of each of them.

The algorithm is depicted below, and it assumes a general reset as initial state. The issue of a signal can either generate an interruption, or be detected by continuously polling the sensors. For the latter alternative, which we have adopted, the order in which the sensors are "visited" depends on their nature. For a uniform generation frequency, a random polling is satisfactory, whereas whether the frequency ranges are disparate, the order can be fixed on the basis of the frequency values.

The next step involves the selection of the PUs that compute data of the same type (field type), which gives the subset U*. If there is at least one available PU of the sought type, these are indexed over the number of items waiting in each queue to be processed. Afterwards, depending on the frequency value of the data token, the maximum charged queue for low frequencies or the minimum charged one for high frequencies is chosen to put in the new data packet. Otherwise, it is sought a spare unit or, if none exists, the search described above is performed over the whole array of PUs. This process implies a reconfiguration of the selected PU, prior to the moment when the data gets computed.

Arbitration Algorithm

```
reset (U, QU)

start:

poll (S[i].data ≠ 0) (i = 1,n) randomly/f(frequency)etc.
get (U*) for (U[j].type = S[i].type)

if (|U*| = 0)
      if (∃U[j] = 0) put (S[i] onto U[j]) goto start
      else U <= U*

order (U*, ascending |QU|)

if (S[i].freq < f_threshold) put (S[i] onto U ≡ U*[jMAX])
else                         put (S[i] onto U ≡ U*[jMIN])

goto start

S=Sensor Array, U=Core Set, QU=Core Associated Queues
```

Fig. 3 illustrates how a data packet is conveyed to the assigned PU. A spare PU is drawn on colourless background. The different colouring of a PUs means it gets configured for the assigned task (having the same colour).

Fig. 2 depicts three special cases: *above* - of a queue getting full, *middle* - of the selection of a spare or *below* - already working PU of another type than that of the incoming data, corresponding to the first if instruction of the algorithm.

3.2 Simulation Results

There are three different situations that a PU can show at a given moment. The first is the standby state, which characterises an idle PU. The other two give a hint whether, after a new task arrives, the PU does not change or it modifies its state. The most expensive case represents a transition from a state to another, which reflects the reconfiguration process that has to be carried out.

The analysis focuses on the main influence due to the number of available PUs for a given set of sensors, variable in number and giving rise to signals at random time moments. The results obtained illustrate that the sensor number preserve the curvature of the graphics, whereas the PU number changes their form. The third factor, the queue number associated to the PUs (we assumed that it is equal for the entire PU array) produces a minor effect on the state variations, under both assumptions, of the algorithm and the random distribution, respectively.

Fig. 4 and Fig. 5 depict the usage of the PUs, when varying their number and applying or not the proposed algorithm or assigning randomly the PUs to the tasks. The abscissa gives the PU number, whereas the ordinate the number of tasks performed by each PU, of a total of 300. (Raising the limit up did not prove any significant change, for the given ratios.) It can be seen that it arrives to saturation (here, for 4 processors), above which the other processors prove needless.

The manner, in which the three situations, which can show up during the task processing of the PUs, are distributed, is reflected in the two consecutive illustrations, Fig. 5 and Fig. 6. The main result is the halving of the reconfiguration number, at the expense of the redistribution of the tasks to PUs already processing one of the same type, or exploiting idle PUs.

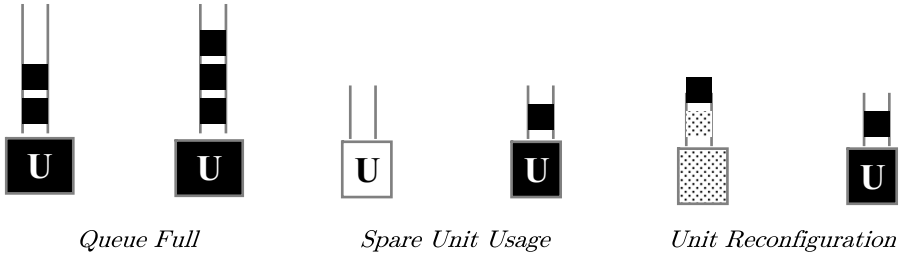


Fig. 2. Three Special Cases of the Queue Control: *left* Filled Queue, *middle* and *right* Selection of a New Unit: Spare or Reconfigured

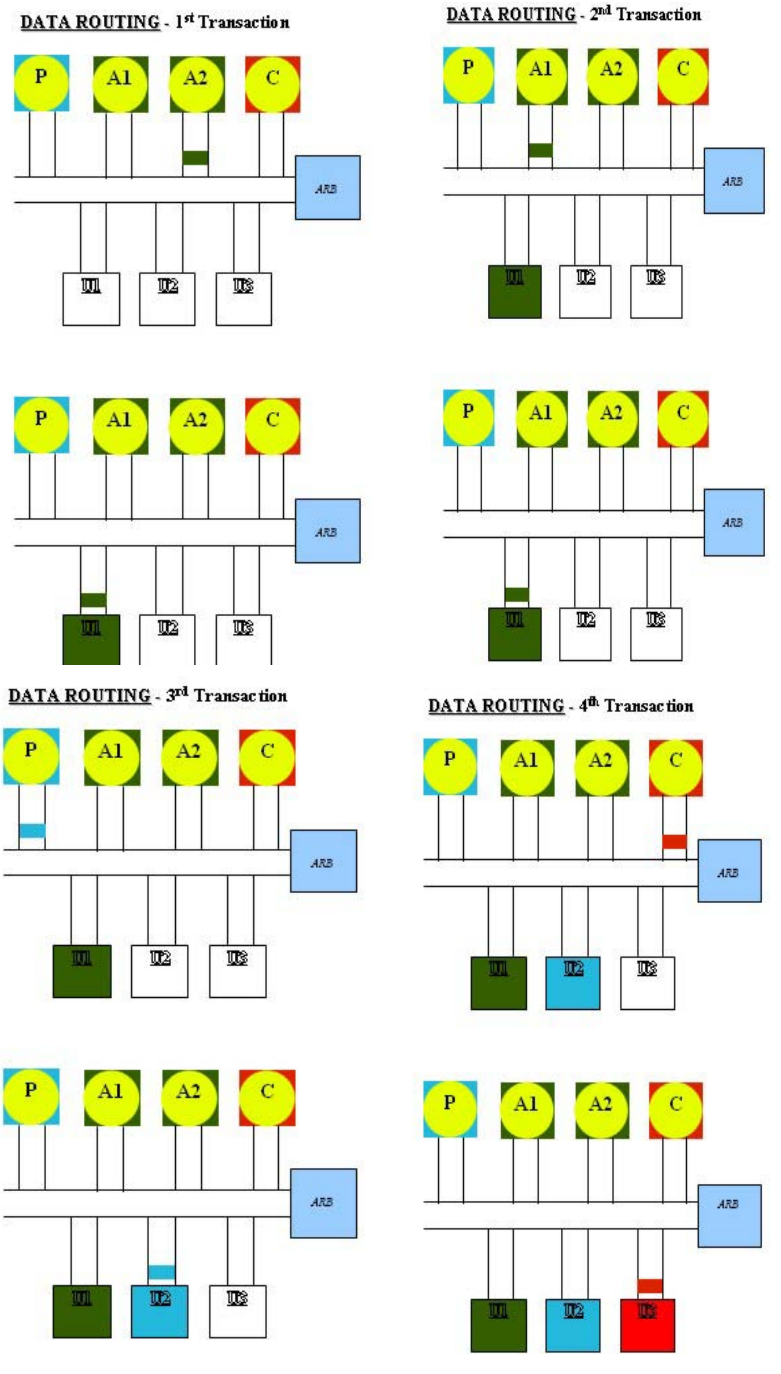


Fig. 3. Data Transaction Sequences from Sensors to PUs depending on the Signal Type

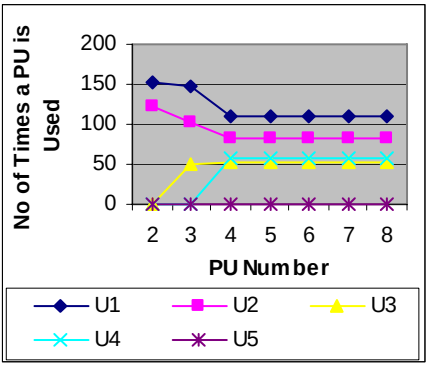


Fig. 4. PU Usage as a Function of PU Number (Algorithm)

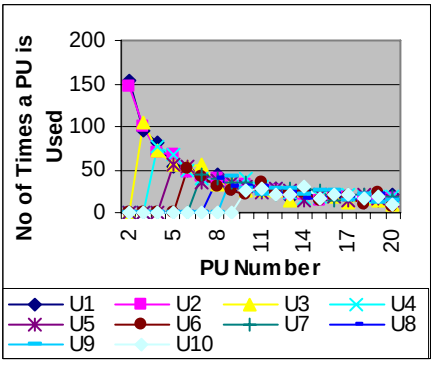


Fig. 5. PU Usage as a Function of PU Number (Random Distribution)

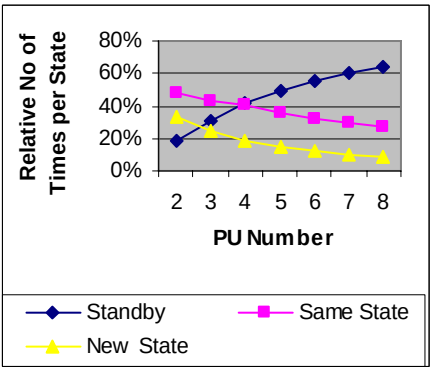


Fig. 6. State Diagram Variation as a Function of PU Number (Algorithm)

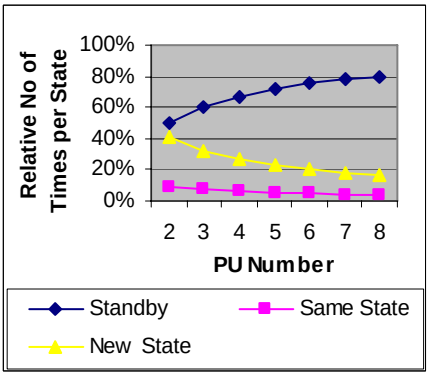


Fig. 7. State Diagram Variation as a Function of PU Number (Random Distribution)

3.3 Network Topologies

The replacement of the traditional bus based structure with more flexible and low power consuming types of interconnection characterises the next step of our system development. On the one hand, the PUs can be generalised, representing different cores, such as: processors, FPGAs, DSPs or IP cores. On the other, the network topology among them covers quite a wide range of alternatives.

Four main strategies are going to be looked at under the magnifying glass. The mesh type [7] (Fig. 8a) has the advantage of regularity. The redundancy of the two router levels of the fat tree scheme [5, 14] (Fig. 8b) confers an increased degree of fault tolerance. The network on chip [4, 5, 6, 8] (Fig. 8c) and point to point (Fig. 8d) architectures are other two options to be analysed.

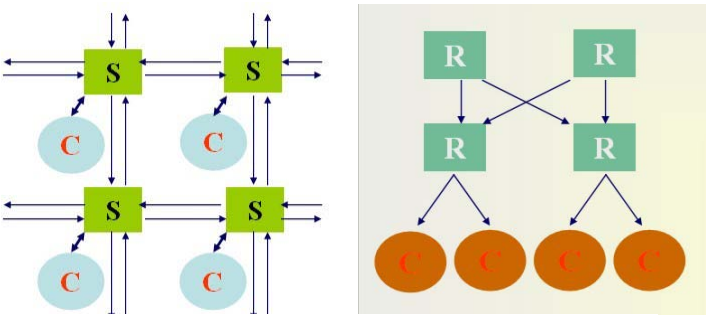


Fig. 8a. and 8b. Different Network Topologies *left* Mesh and *right* Fat Tree

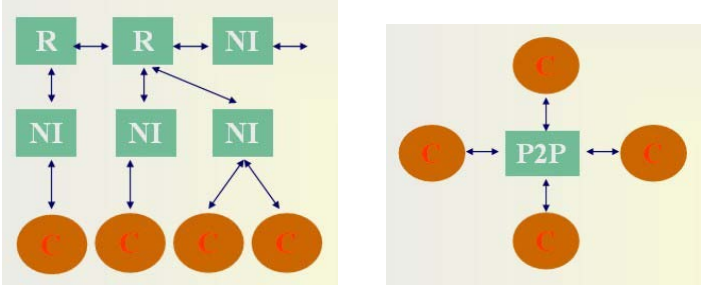


Fig. 8c. and 8d. Different Network Topologies *left* Network on Chip and *right* Point to Point

4 Conclusions

We develop a task allocation technique for the signals generated by a set of heterogeneous sensors and sent to a processor unit array, so that the energy consumed be reduced. The main goals are to lower the number of state transitions of the PUs involved, and to increase the period during which a PU maintains in the standby state. A PU is programmed to perform various computation tasks, one at a time, and it has the capability of switching from one to another, depending on the incoming data type. Data processing speed increase and data dropping minimisation are other two topics addressed in the present article, the fulfilment of which is insured by the algorithm performed by the arbitration unit, on the way from sensors to PUs, as well as by the usage of queues for each PU. A small number of PUs consuming less energy is the main goal of the application of the proposed algorithm, mainly due to the lowering of the reconfiguration actions, that, on the other hand, ensure higher computation flexibility. Eventually, the issue frequency of the outgoing sensor signals conditions their allocation to the corresponding queue attached to every PU. In the next phase, the implications onto the energy consumption of various interconnection topologies among the PUs will be analysed.

References

1. Kirianaki, N., et al.: Data Acquisition and Signal Processing for Smart Sensors. John Wiley & Sons, Chichester (2001)
2. Yurish, S. (ed.): Smart Sensors and MEMs. Kluwer Academic, Dordrecht (2005)
3. Bhat, S.: Energy Models for Network-on-Chip Components, M.Sc. Thesis, Technische Universiteit Eindhoven (2005)
4. Wiklund, D., Sathe, S., Liu, D.: Network on chip simulations for benchmarking. In: Proceedings of the International Workshop on SoC for Real-Time Applications, Banff - Canada (2004)
5. Benini, L., DeMicheli, G.: Networks on Chips: A New SoC Paradigm. Computer, vol. 7078 (January 2002)
6. Kumar, S., et al.: A Network on Chip Architecture and Design Methodology. In: Proceedings of IEEE Computer Society Annual Symposium on VLSI, vol. 117, IEEE Computer Society Press, Los Alamitos (2002)
7. Zeferino, C., Creutz, M., et al.: A Study on Communication Issues for Systems-on-Chip. In: Proceedings of Integrated Circuits and Systems Design (SBCCI), vol. 121 (2002)
8. Vermeulen, B., et al.: Bringing Communication Networks on a Chip: Test and Verification Implications. Communications Magazine, IEEE 41(9), 74–81 (2003)
9. Brockmeyer., et al.: Parametrizable Behavioral IP Module for a Data-Localized Low-Power FFT. In: Proc. IEEE Workshop on Signal Processing Systems (SIPS), Taiwan, pp. 635–644 (1999)
10. Maro, R., et al.: Dynamically Reconfiguring Processor Resources to Reduce Power Consumption in High-Performance Processors, Workshop on Power-Aware Computer Systems, Cambridge, MA (2000)
11. Ienne, P., Pozzi, L., Vuletic, M.: On the Limits of Automatic Processor Specialisation by Mapping Dataflow Sections on Ad-hoc Functional Units, Technical Report 01/376, EPFL, Lausanne (2001)
12. Bai, Y., Bahar, R.I.: A Low-Power In-Order/Out-of-Order Issue Queue. ACM Trans. on Architecture and Code Optimization 1(2), 152–179 (2004)
13. Sinha, A., Chandrakasan, A.: Energy Efficient Real-Time Scheduling. In: Proceedings of the 2001 IEEE/ACM International Conference on Computer-Aided Design, San José, CA, pp. 458–463 (2001)
14. Guerrier, P., Greiner, A.: A Generic Architecture for On-Chip Packet-Switched Interconnections, DATE 2000, 250 (2000)

Mini Robots for Soccer

Man-Wook Han and Peter Kopacek

Intelligent Handling and Robotics (IHRT),
Vienna University of Technology
Favoritenstrasse 9-11, A-1040, Vienna, Austria
{han, kopacek}@ihrt.tuwien.ac.at

Abstract. Robot soccer was introduced with the purpose to develop intelligent cooperative multi-robot (agents) systems (MAS). From the scientific point of view a soccer robot is an intelligent, autonomous agent, carrying out tasks together with other agents in a cooperative, coordinated and communicative way. Robot soccer is a good test bed for the development of MAS for current and future industrial applications. The robots in a team have a common goal – to kick the ball in the opponent goal and to avoid goals against the own team. The cooperation and coordination of actions by means of communication are necessary. In this contribution the development of three generations of mini robots for robot soccer, in the FIRA categories Miro- and Narosot, are described. Finally "Roby Space" as a "spin-off" of robot soccer is presented.

Keywords: Robotics, Mechatronics, Robot Swarms, Edutainment.

1 Introduction

The robot-soccer system (FIRA Miro- and Narosot) works as follows: A camera approximately 2m over the playground delivers 60-100 pictures/second to the host computer. With color information on the robots, the vision software calculates the position and the orientation of the robots and the ball. With this information the host computer calculates motion commands based on the implemented game strategy and send these commands wireless to the robots.

Already MIcro-RObot SOccer Tournaments (MiroSot 96, MiroSot 97) were organized by FIRA (Federation of International Robot-soccer Association) in Korea. In 1998 the Micro-Robot World Cup Soccer Tournament took place in Paris in conjunction with the FIFA World Cup 1998. Over 50 teams from all over the world participated in MiroSot '98. In the following years world cups were organized in Campinas 1999, Rockhampton 2000, Beijing 2001, Korea 2002, Vienna 2003, Busan 2004, Singapore 2005 and Dortmund 2006.

In Europe first demonstration games were organised at Vienna University of Technology in September 1997 and the first FIRA European Cup took place 1998 in Vienna. 4 teams from Korea, Switzerland, Spain and Austria attended this competition.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1009 – 1016, 2007.
© Springer-Verlag Berlin Heidelberg 2007

A team consists of three robots and the games were carried out on a playground of 1.5m x 1.2m. The image processing system was completely in analogue and operated only with 30 frames/sec.

Meanwhile there are different categories according to the size of and playgrounds and the number of playing robots. In FIRA Miro- and Narosot five or eleven robots build a team. The dimensions of the related playgrounds are: 220 x 180cm and 400 x 280cm. In the category of large league (eleven vs. eleven) most of the teams are using two cameras with a speed of up to 200 frames/second.

One of the latest developments is the extended (e-) middle league: 5 vs. 5 robots on a playground of 7 vs. 7 (280 x 220cm).

2 The IHRT Soccer Robot Family

Until 1998 the Team AUSTRO from Vienna University of Technology played with robots from Korea. At that time the robots were very slow (approx. 4 km/h) with a low acceleration ($3m/s^2$) and were not constructed in a robust mechanical construction. In 1997 the development of our first soccer robot "Roby Go" was initiated. At IHRT three generations of soccer robots "Roby Go", "Roby-Run" and "Roby-Speed" as well as "Roby Naro" were developed [1].

It is important to develop fast and robust mobile mini robots. The robots have to be small in size. The robots in the category MiroSot may be a cube with a side-length of 75 mm. In the NaroSot category the size of the robots may not exceed 40 mm x 40 mm x 50 mm (l x w x h). For the competition the following features were necessary:

- Speed up to 3.5 m/s and acceleration more than $8 m/s^2$
- Robust
- Tolerance of position less than ±0.4cm/m
- Signal (LED or acoustic) of robot's internal states
- Fast and easy batteries reloading
- Low centre of gravity
- Possibility for driving the ball
- Roller bearings and a single stage gear
- Modular and open software architecture

Table 1. Specifications of the IHRT – robots

	Roby Go	**Roby Run**	**Roby Speed**
Microcontroller	C167 (Infineon)	C167 (Infineon)	XC167 (Infineon)
Communication Module	RF frequency : 418 or 433 MHz	RF frequency : 418,433 or 869 MHz	RF frequency : 418,433, 869 or 914 MHz
Software Controller	PID, Neuro PID or advanced algorithms Sampling Time 1 ms	PID, Neuro PID or advanced algorithms Sampling Time 1 ms	PID, Neuro PID or advanced algorithms Sampling Time 1 ms
Dimensions in mm (Length x Width x Height)	75 x 75 x 75	75 x 75 x 75	75 x 75 x 46

Table 1. (*continued***)**

Motors	Faulhaber 2224 06SR	Faulhaber 2224 06SR	Faulhaber 2224 06SR
Sensors		Acceleration sensor	Acceleration sensor
Power	NiMH 9 Cells, 1.2V, 550mAh	NiMH 9 Cells, 1.2V, 700mAh	LiIo 11.1V 1400 mAh
Max. Speed (m/s)	2.54	3	3.9
Max. Acceleration (m/s^2)	2.54	6	10
Main Computer	Pentium IV 1.5 GHz OS: Windows 98	Pentium IV 2.8 GHz OS: Windows 2000	Centrino 2.26 GHz OS: Windows XP
Vision System	Analogue, 30 half pictures/s	Digital, 60 full pictures/s	Digital, 200 full pictures/s

2.1 Roby Go

"Roby Go" (Fig. 1) is a two wheel-driven mini mobile robot, distinguished by its simple, compact and modular construction. It has two PWM controlled DC motors. The microcontroller C167 is used which can be programmed by the serial port and has a CAN bus interface. Therefore it is possible to connect several microcontroller boards for different tasks. Additional sensors can be easily attached in the future. The electronic part has a modular and open architecture and consists of two boards, one board is responsible for power electronic and communication and one for microcontroller. The control software has an open architecture.

2.2 Roby Run

In 2002 "Roby Run" (Fig. 2) was ready for application. The major improvements to "Roby Go" were a more compact mechanical construction, a miniaturized electronics together with a more powerful microcontroller, the implementation of acceleration sensor to avoid slippage, the lower centre of gravity, and the secure communication between robots and the host computer. The speed and the acceleration were higher (Table 1). "Roby Run" was much more robust than "Roby Go".

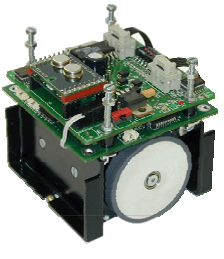


Fig. 1. "Roby-Go"

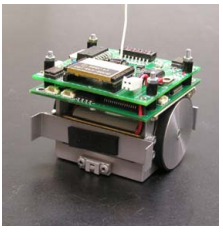


Fig. 2. " Roby Run "

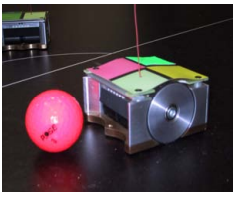


Fig. 3. Roby Speed

2.3 Roby Speed

In the last years the speed and acceleration of the robots increases dramatically. Therefore “Roby Speed” (Fig. 3) has a maximum speed of 14km/h with an acceleration of 10m/s². Further improvements to “Roby Run” are:

- the robust and compact mechanical design
- the use of Lithium-Ion rechargeable batteries
- infrared interface
- the use of special tire for two wheels
- the miniaturized electronics: the two PCB`s (7.4 x 7.4cm) are now reduced to one (3.2 x 3.2 cm) and fully integrated in the body.

2.4 Roby Naro

For special applications a smaller robot with the dimensions 4 x 4 x 5cm, “Roby Naro” (Fig. 4), was developed [2]. Sometimes a robot has to reach positions in a very “narrow” environment.

In addition the robot plays soccer in the FIRA Category “NaroSot”. Lot of so called “nano” robots are worldwide available – mostly prototypes. They are far away from nano dimensions. The sizes are between some mm to cm, but some of the parts are in currently in micro or probably in the nearest future nano dimensions. Today we have two categories in development:

- Fully autonomous: On-board Nanocomputer for control
- „Insect robots“: like ants, bees…..mostly in swarms controlled by a host computer.

“Roby-Naro” (Fig. 4 and 5) will be used for other applications in the future. Therefore additional features are:

- Robust and closed design for protection of all electrical and mechanical components
- Drives with high power in relation to weight and size of the robot
- Appropriate ball guidance.

The technical details are collected in Table 2.

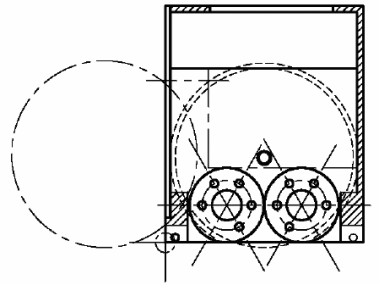


Fig. 4. Design draft of “Roby-Naro” [3]

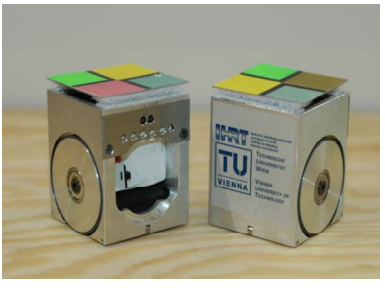


Fig. 5. Roby Naro

Table 2. Technical data of Roby Naro

Dimensions:		
Length:	40	mm
Width:	40	mm
Height without electronic:	49	mm
Height with electronic:	49	mm
Wheel diameter:	36	mm
Wheel width:	4	mm
Wheel base:	36	mm
Weight of robot:		
Without battery pack:	100	g
With battery pack:	150	g
Specification of PWM controlled DC Motors:		
Minimotor Type:	1524 06 SR	
Output power:	1.70	W
Speed up to:	10000	rpm
Stall torque:	6.68	mNm
Encoder resolution:	512	ppr
Single stage gear(good efficiency):		
Gear ratio:	112:9	
Movement data:		
Maximum speed:	2,3	m/s
Maximum acceleration:	3	m/s^2

3 Hard- and Software

In the following more details on the robots, “Roby Speed” and “Roby Naro” will be given.

3.1 Hardware

The hardware control unit (Fig. 6 and 7) is identical for both robots. With an area of 36 x 36 mm the board fills only 1/8 of the reference volume and consists of the following components:

- XC167 microcontroller from Infineon with internal RAM (8 kByte) and Flash (128 kByte)
- Power supply by switching regulators with high efficiency
- High speed dual full bridge motor driver
- Infrared transmission module
- Bi-directional radio module for the frequencies 433, 869 or 914 MHz
- Status indication by six bright LED's in different colors
- Serial synchronous interface for communication with further modules, e.g. XScale board [3].

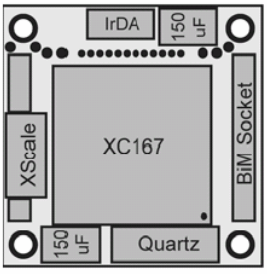


Fig. 6. Top view of the electronic board

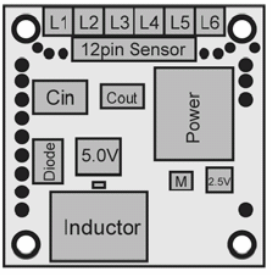


Fig. 7. Bottom view of the electronic board

The printed circuit board (Fig. 6 and 7) consists of 4-layers. Because of the high density of parts the PCB contains holes with diameters of only 0.15 mm. The DC - motors are controlled by PWM signals generated by the microcontroller. Due to the fact, that the motors need much more electric current than the microcontroller, a dual full bridge driver is installed. For the radio communication a special module is used, which can be plugged in this board. For selecting a robot - remember up to eleven robots are playing in a team - an infrared transceiver is integrated. It allows a flexible, wireless adjustment of all control settings by communication software running on a PDA (Personal Digital Assistant). Additionally six LED's are implemented to check whether

- the robot is turned on,
- status of the power is and
- Radio communication works properly.

The robots are equipped with a piezo speaker unit for further communication and entertainment. The robot is driven by 11.1V lithium-ion battery packs. A high efficient switching regulator transforms this voltage to the 5V and respectively 2,5V microcontroller level.

For the vision system there are three main problems:

- A sampling time of even 20 ms is too long to control a fast moving robot driven by DC-motors.
- The time between taking a picture of the playground and the reaction time of the robot is 20ms.
- The vision system is inaccurate.

Because of this delay time some prediction algorithms (e.g. Kalman filters…..) are implemented.

The acceleration of the robot is measured by two biaxial sensors of the type ADXL203 from Analog Devices. They support the local speed control and allow the detection of slip at the drive wheels of the robot [5].

3.2 Software

The software package consists of three main parts:

- vision system
- strategy-module
- communication-module

For the vision system one or two colour-CCD-cameras are used. Based on the information from the cameras the main computer is responsible for image processing, the generation of commands for robots according to the chosen strategy, and the communication with other robots. To find one of the best movements a "heuristic search strategy" was implemented. For each robot different movements will be calculated by the host computer and the best will be sent to and executed by the robots.

3.3 Vision System

The speed of the vision system is very important. There is a time delay (approx. 20 ms) between taking a picture, image processing, generation of motion commands, sending these commands, receiving the motion commands and movement of the robot. In this time interval between taking a picture and submit the movement command to the robot the robot is in another position. Assume that the robot moves at a maximum speed of 3 m/s and the time delay takes 20 milliseconds, the difference between robot's position when the camera takes a picture and robot's position when the robot starts to move is approximately 6cm. Furthermore the time between new data provided by the vision system is not constant. Therefore it is necessary to have a fast as well as an exactly working vision system.

4 Roby Space

As the spin off from robot soccer, IHRT became a European Space Agency - ESA project [6]. In the frame work of this project a small mobile robot (10 cm x 10 cm x 5cm) which is able to crawl on the net structure 200 km over the earth. "Roby Space" was developed. The requirements were:

- the maximum dimension 10x10x5cm
- light weight (less than 1 kg)
- on-board power supply for approximately 10 min.
- equipped with a camera sending pictures to the earth
- wireless communication with the mother satellite by Bluetooth or similar
- free movement on the mesh
- mechanical and electronic robustness against vibration and shock

Because of the know-how getting from robot soccer – approximately 75% of the hard- and software was from "Roby Speed" - the project was finished successfully in January 2006 by a test launch with a Japanese rocket [6].

5 Conclusion

In this paper the development of a family of mobile mini (naro) robot is described. The team AUSTRO won with these robots some categories in FIRA World Cups as well as European Cups.

Especially in NaroSot only few teams can participate in the competition. A problem was the stability of the Narorobots. They can only move with a reduced speed.

Future works are the improvement of the intelligent and autonomous behavior by means of adding visual sensors and the realization of full communication between robots to carry out their tasks in coordinated as well as cooperative ways.

Industrial applications are in progress.

References

1. Kopacek, P.R.: Past, Present and Future. In: Proceedings of ISR 2006, Munich, Germany 2006 p. 153 (2006)
2. Kopacek, P., Han, M.-W., Putz, B., Schierer, E., Würzl, M.: Narosot – Nanotechnology in Robot Soccer. In: INCOM 2006. Preprints of the Conference on Information Systems, Control and Interoperability, Saint-Etienne, France, pp. 193–198 (2006)
3. Putz, B.: Development of the new Soccer Robots "Roby-Speed" and "Roby-Naro". In: ELH 2004. Proceedings of CLAWAR/EURON/IARP Workshop on Robots in Entertainment, Leisure and Hobby, Vienna, Austria, pp. 47–52 (2004)
4. Würzl, M.: A contribution to robotics especially in the field of expert- and multi agent systems, PhD thesis, Intelligent Handling and Robotics, Vienna University of Technology (in German, 2005)
5. Putz, B., Han, M.W., Kopacek, P.: Implementation of acceleration sensors improving dynamic behaviour of a mobile mini robot. In: IAV 2004. Preprints of the 5th IFAC/EURON Conference on Intelligent Autonomous Vehicles (2004) TP-6-5
6. Han, M.W.: Mobile Mini-Robots for Space Applications. In: CLAWAR 2005. International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, pp. 1037–1044. Springer, New York (2006)

Embedded Robotic Solution: Integrating Robotics Interfaces with a High-Level CPU in a System-on-a-Chip

Florian Eibensteiner[1], Rainer Findenig[1], Jürgen Tossold[1], Wilfried Kubinger[2], Josef Langer[1], and Markus Pfaff[1]

[1] University of Applied Sciences of Upper Austria, Austria
[2] Austrian Research Centers GmbH - ARC, Austria
florian.eibensteiner@fh-hagenberg.at

Abstract. In today's robotic systems, on one hand, time-critical tasks have to be executed, like the management of several different sensors and actors, while, on the other hand, highly sophisticated jobs, for example positioning systems, node networks, or web-based user interfaces, need to be performed in parallel.

Conventional solutions, which are mostly deploying an high-end microcontroller as well as one or more separate integrated circuits which are in charge of the sensors and actors, have significant disadvantages: they are inflexible, need more PCB place, and consume more power.

System-on-a-Chip (SoC) based approaches minimize the complexity and PCB place as well as power consumption, and therefore the costs, of such systems. Since all functionality can be included in one single chip, the PCB's size and the system's power dissipation can be reduced.

Keywords: System-on-a-Chip, Robotics, RTOS, hard real-time.

1 Introduction

In this paper we present a System-on-a-Chip based solution for an integrated hardware platform controlling small robots.

Today's high-end embedded robotics applications require the handling of two, more or less mutually exclusive, types of tasks: hard real-time tasks need to be executed in parallel with less time-critical, but highly sophisticated jobs. Those include networking capabilities for communication amongst several autonomous robots, web-based and remote configuration for the user, as well as the control and synchronization of advanced workflows and plenty of other applications. The real-time tasks the system has to handle are obvious: sensor inputs have to be processed and acted upon in a specified amount of time, and commands for actuators like servos have very strict timing constraints.

2 Related Work

Yaakob et al. presented an SoC-based approach for industrial robotic controller design based on a 6502 processor. They developed the SoC as an alternative

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1017–1024, 2007.
© Springer-Verlag Berlin Heidelberg 2007

electronics solution to an educational robot [1]. It is obvious, though, that the 6502 processor's potential is too little to run a full-featured operating system which incorporates network and multitasking capabilities.

Ahmet Bindal et al. presented a synopsis of an undergraduate SoC course for computer engineering students in which they deployed an Altera FPGA with an integrated ARM core [2]. This obviously makes the system dependent on the FPGA's manufacturer and, moreover, the core itself is a closed-source solution.

Our approach is based on an open-source CPU model, namely the Leon3 core from Gaisler Research [3], and therefore offers a large advantage, not only for educational use but also for open source commercial applications.

3 Problem Analysis: Tasks of an Embedded Robotic System

Embedded robotic solutions need to execute several tasks with different levels of sophistication and real-time demands concurrently.

3.1 Hard Real-Time Tasks: Servo Controlling

Controlling a simple servo motor is an example of a hard real-time task; the findings in this section are, nevertheless, in no way limited to servo motors and, obviously, can be applied to other time-critical jobs, too.

Tremors and Accuracy: In the first application of our approach we are deploying servo motors which are controlled through a pulse width modulation with a frequency of 50 Hz (see figure 1):

$$\begin{aligned} \alpha = 0^\circ &\Longleftrightarrow t_p = 1.5 \text{ ms}, \\ \alpha = -50^\circ &\Longleftrightarrow t_p = 2.1 \text{ ms}, \\ \alpha = 50^\circ &\Longleftrightarrow t_p = 0.9 \text{ ms}, \end{aligned} \tag{1}$$

where t_p is the high time of the servo pulse and α is the angle of the servo.

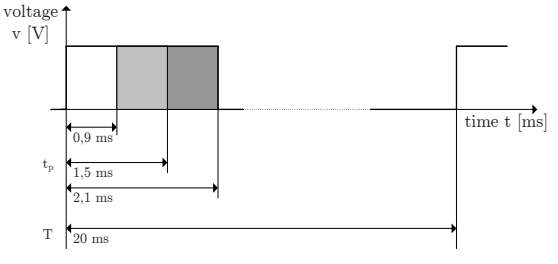


Fig. 1. Pulse width modulated servo signal

These values are, of course, dependent on the type of servo motors that are used in the design, but the calculation itself remains unchanged for other motors.

The relations given in equation 1 lead to the following observation (t_0 is a constant offset and K describes the relation between t_p and α):

$$\begin{aligned} \alpha &\sim (t_p - t_0 - 1.5 \text{ ms}) \\ \alpha &= K \cdot (t_p - t_0 - 1.5 \text{ ms}) \\ K &= \frac{\alpha}{(t_p - t_0 - 1.5 \text{ ms})} \end{aligned} \tag{2}$$

Inserting the known extrema for α and t_p for a given servo motor, we can calculate K using equation 2[1].

In order to calculate the needed pulse time for a desired angle α, we can use the following equation:

$$t_p = \frac{\alpha}{K} + (1.5 \text{ ms} + t_0)\,. \tag{3}$$

This makes it obvious that it is imperative that pulse time jitter be prevented, or otherwise exact positioning is futile because of the induced tremor: the high-time of the pulse directly effects the servo's angle and therefore jitter in time results in small movements of the servo even if the angle is to be held constant.

On first glance, it seems simple to meet the above constraints and still keep jitter to a minimum, even with a software based solution—after all, the cycle time of the pulse width modulation is 20 ms and the high-time of the pulse conventionally ranges between 0.9 ms and 2.1 ms. These are time ranges that can be effortlessly handled in software, even with low-cost microcontrollers. Nonetheless, remember that this allows control of nothing but *one* servo; today's sophisticated robots may have dozens of them, all of which need to be controlled in parallel. Moreover, the scope of this paper is not a platform that only drives motors, but also other, much more sophisticated and computationally intensive tasks.

When looking at the necessary accuracy of the pulse high time to achieve a given accuracy of the servo, it becomes obvious, though, that even for one servo, a tremor cannot be prevented completely:

$$t_{Interval} = \frac{1}{K} \cdot accuracy, \tag{4}$$

which, for an assumed K of $0.083°/\mu$s and a desired accuracy of $0.1°$, leads to the maximum jitter time being

$$t_{Interval} = \frac{1\ \mu\text{s}}{0.083°} \cdot 0.1° \approx 1.2\ \mu\text{s}. \tag{5}$$

1.2 μs are well within the interrupt jitter range for conventional microcontrollers, as they correspond to 10 cycles at 8 MHz (which is a typical frequency for e. g. an AVR) and 60 cycles at 50 MHz (which is a typical frequency for

[1] This is an approximate value we determined for the servos used in our application. An in-depth discussion of K will be presented in subsection "Nonlinearity Correction".

e. g. an ARM processor in a low-power environment). Therefore, the given accuracy is unreachable without either deploying a processor with a very low interrupt latency or increasing the operating frequency of the processor significantly.

Nonlinearity Correction: Real servo motors are highly nonlinear, resulting from fabrication tolerances and the abrasion and the load on the drive axle. Therefore, equation 3 is not completely correct. In fact, K is not a constant, as suggested in this formula, but depends on the current angle α:

$$t_p = \frac{\alpha}{K(\alpha)} + 1.5 \text{ ms} + t_0. \tag{6}$$

Finding a formula that models K tightly is hardly possible and in any case very tiring, since K differs greatly even between servos of the same type. The only feasible way is sweeping the pulse width and measuring the exact position of the servo, thereby effectively creating a characteristic curve for each servo. This characteristic curve has to be compensated for each servo individually, and in real time, every time a new pulse length is calculated for the servo.

A typical characteristic curve is shown in figure 2. The complexity of such a curve makes it hardly sensible to completely store it in a look-up table: it would be tremendously big; note that the resolution provided in figure 2 is twelve bit, therefore two bytes are needed for every value stored in the table!

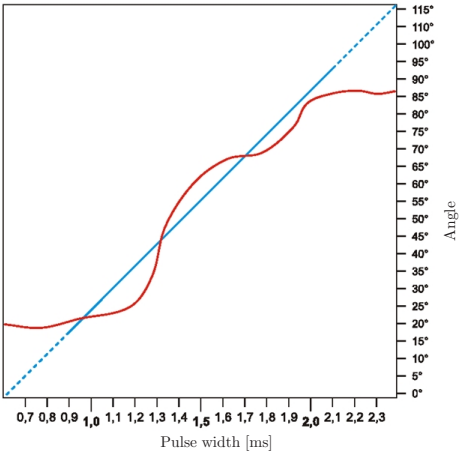


Fig. 2. Characteristic curve of a servo

The other option is storing only a few supporting points and interpolating between those. This is of course feasible, but this adds high computational overhead for the software—remember that this interpolation needs to be done every single time a new pulse width is calculated for a servo and that the supporting points can differ significantly between servos so that the result cannot be reused for other servos.

3.2 Control and Communication Tasks

Most of today's highly demanding applications require networking capabilities which range from different industrial buses like LIN, CAN, and FlexRay over conventional, mostly preexisting systems like Ethernet to wireless solutions like wireless LAN, ZigBee, or others. For a great number of widely used network standards, complex protocol stacks need to be implemented: for non-standard processors they mostly need to be ported or even implemented from scratch which is a highly time-consuming and error-prone task. Moreover, a controller needs to be employed that either features a hardware core to control the lower layers of the network or the network has to be simulated in software ("bit banging") which again increases the application's load and real-time requirements.

For autonomous robots, not only communication, but also positioning systems are required. This includes, but is not limited to, using GPS positioning in combination with either a stored map or a map that is generated on the fly using sensors. Both options are computationally intensive and require high-level algorithms which can hardly be implemented in a small, low-cost microcontroller.

The development of all of the above can be greatly eased by using an embedded operating system that not only provides multitasking capabilities but also well-tested libraries and drivers [6]. Obviously, though, a sophisticated operating system needs a powerful microprocessor to run sensibly. In addition to facilitating the development, an operating system can also greatly increase portability and reusability between different applications, thereby increasing code quality and decreasing time to market.

4 Embedded Robotic Solution: The SoC Approach

In the following section, we describe in detail how the basically conflicting needs of such an application can be met using an SoC and present our approach to address the problems described before.

The first step in creating a system-on-a-chip is partitioning the system into hardware and software. Our system is based on an FPGA which allows for great flexibility in partitioning, even in later design phases. Basically, the idea of our approach is to execute the time-critical jobs in hardware where the inherent parallelism can be exploited. Thereby, the load of the processor can be reduced, while, at the same time, the computational capacity of the system can be increased.

4.1 Hardware

An overview of the system is presented in figure 3. Communication takes place over the AMBA AHB/APB bus which is a industry standard bus and therefore allows easy connection of third-party IP cores [7].

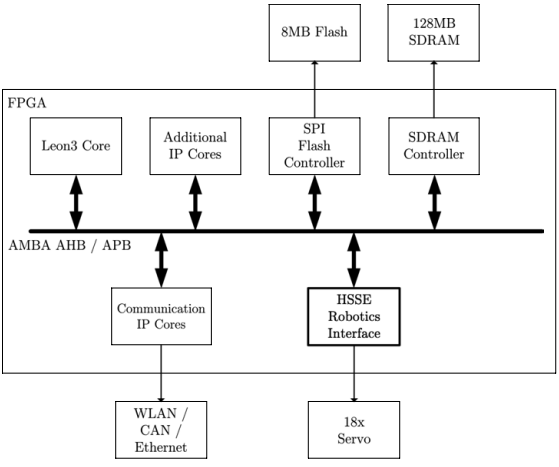


Fig. 3. Overview of the system-on-a-chip

Processor: The Leon3 processor from Gaisler research forms the core of our system-on-a-chip. It is a softcore provided open-source under the terms of the GNU General Public License (GPL) [5,6]. This is especially appealing for universities, but of course also for companies that either develop open-source solutions or want to try the core before buying it under a commercial license.

The processor implements a SPARC V8 architecture and features an MMU as well as several ready-to-use IP cores needed to use it in advanced systems, including, but not limited to, an 10/100/1000 MBit Ethernet MAC, SDRAM and DDRAM controllers, a PCI interface, and CAN and LIN bus controllers.

The benefits when using a softcore processor on an FPGA based system-on-a-chip are obvious: the periphery and the processor can be tailored to every single application. This includes, of course, the possibility to add and remove IP cores as needed, but can also go as far as adapting the processor itself, be it simple things like tuning the cache size to get the best performance or complex additions like enhancing its instruction set or adding co-processors.

HSSE Robotics Core: A specialized IP core was developed to handle real-time critical tasks (see figure 4 for a block diagram). It is designed to handle up to 24 servo motors concurrently, scaling the pulse lengths for each of those servos with an accuracy of the core's cycle time—therefore, at e. g. 25 MHz, we can achieve an accuracy of $\frac{1}{25\ \mathrm{MHz}} = 40$ ns. This, in turn, corresponds to an accuracy of the servo's angle (see equation 4) of approximately 0.003°, which is far beyond what current servos are able to offer given the mechanical problems we described when we outlined the nonlinearity of K.

In addition to providing this accuracy for each servo, even when used in parallel, the core features a linearization unit which allows to store a characteristic curve inside the HSSE Robotics Core, specified by nine supporting points, for each of those servos. The needed interpolation (which is done linearly) is done

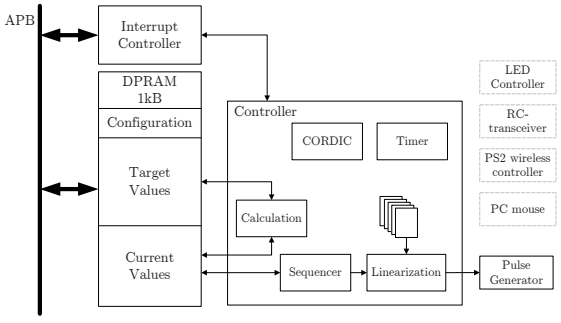


Fig. 4. Overview of HSSE robotics core

completely in hardware, therefore after the initialization of the internal tables storing the supporting points no additional software effort is necessary to fully linearize the characteristic curve.

An integrated sequencer for each servo allows the controller to program complex movement sequences in advance which are then executed independently by the robotics core.

Optional hardware components include support for various input systems, for example, a controller for an RC-transceiver and a Playstation 2 wireless controller, in order to allow for great flexibility in the way the robot is controlled.

4.2 Software

As suggested before, we are using an embedded operating system to facilitate the development of robotics solutions that need to deal with sophisticated and complex tasks. The operating system of choice, in our application, was eCos [4], which not only provides basic capabilities such as multithreading but also a vast amount of thoroughly tested libraries. It is licensed under the GPL, which means that it can be used without paying license fees for development tools or any royalties for the devices shipped with it.

eCos provides a great amount of device drivers, protocol stacks and libraries and allows for package-based configuration, which means that the developer can simply use a graphical user interface to configure the operating system's internals as well as which drivers and libraries are provided to the application.

An overview of the system is provided in figure 5. It is dissected into three main tasks: a control thread, an actuator thread, a sensor thread, and a thread to handle the communication. The actuator thread is only used to supply the robotics core with new data, all calculation is done by the core. Data transfer from the core to the control thread is done by the sensor thread; again, no calculation needs to be done in software.

The interface to the outside world is handled by the communication thread, which is able to use protocol stacks and other libraries of the operating system.

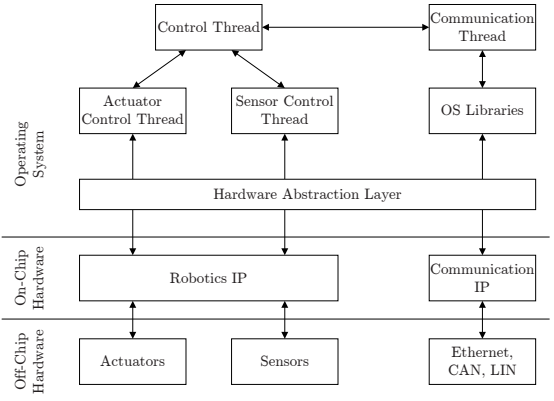


Fig. 5. Overview of the complete robotics system

5 Summary

In this paper we presented an approach coping with the basically conflicting needs of today's robotics systems: while, on one hand, hard real-time tasks need to be executed and deadlines have to be met, on the other hand, complex calculations and elaborate control tasks need to be performed in parallel.

The SoC approach presented allows for easy combination of both those types of tasks: real-time critical parts are implemented in hardware which makes reaching the given deadlines easy and, additionally, allows for a much higher accuracy. A high-level CPU is utilized so the developer does not need need to reinvent the wheel but can deploy a sophisticated operating system, which, in our case, was eCos.

References

1. Yaakob, W.F., et al.: System-on-a-Chip Approach for Industrial Robotic Controller Design. In: ICSE 2000 Proceedings (November 2000)
2. Bindal, A., et al.: An Undergraduate System-on-Chip (SoC) Course for Computer Engineering Students. IEEE Transactions on Education (May 2005)
3. Gaisler Research AB: Gaisler Research Homepage (October 2006), `http://www.gaisler.com`
4. embedded Configurable operating system (December 2006), `http://ecos.sourceware.org/`
5. Free Software Foundation, Inc.: GNU General Public License (December 2006), `http://www.gnu.org/copyleft/gpl.html`
6. Findenig, R., Priewasser, R., Eibensteiner, F., Pfaff, M.: Einsatz von Embedded Linux auf dem Leon3 am Beispiel eines Hexapod-Roboters. Tagungsband Austrochip 2006, TU-Wien and FH-Technikum-Wien (October 2006)
7. Keating, M., Bricaud, P.: Reuse Methodology Manual for System-on-a-Chip Designs. 3300 AH. Kluwer Academic Publishing Group, Dodrecht, Netherlands (2005)

An Embedded Vision Sensor for Robot Soccer

Wilfried Kubinger[1], Franz Rinnerthaler[1], Christoph Sulzbachner[1], Josef Langer[2], and Martin Humenberger[1]

[1] Austrian Research Centers GmbH – ARC, A-1220 Vienna, Austria
[2] University of Applied Sciences Hagenberg, A-4232 Hagenberg, Austria
wilfried.kubinger@arcs.ac.at

Abstract. We present an embedded vision sensor to be used for robot soccer in the MiroSot league of the Federation of the International Robotsoccer Association. The vision sensor is based on a DSP/FPGA co-processor system and uses a FireWire-camera. The vision algorithms are specifically designed to optimally utilize the resources of the embedded system. The achieved embedded vision sensor is able to work at full camera framerate (60fps) with full image resolution (640x480 pixels) without the need of any resources of the host computer.

Keywords: Embedded Vision System, Robot Soccer, Digital Signal Processor (DSP), Field-programmable Gate Array (FPGA).

1 Introduction

Computer vision is a compute-intensive task. Real-time vision systems usually guarantee low reaction times, and must therefore have a high processing rate. Application areas include, for example, automotive engineering or mobile robotics, where processing rates between 10 and 30 frames per second (fps) present the lower limit [1].

In the so called MiroSot league of the Federation of the International Robotsoccer Association [2], computer vision is used to detect, identify and separate play ground and robots of both teams. Since these tasks are very time consuming, significant portions of the available resources of the host computer are occupied. Just a minor part of the processing resources is usually dedicated to other tasks, like interaction with the human operator, control of robots or executing the strategy.

To overcome this problem, we offloaded the whole vision-related work to an external embedded vision system. The advantage of this decomposition is the possibility to design and test all parts of the system independently. Furthermore, since the vision algorithms are executed on a dedicated vision system, a high-end computer for executing the other tasks is no longer needed.

2 Related Work

Autonomous systems are mostly based on embedded systems, due to the limited robot sizes. Many of theses systems are based on digital signal processors

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1025–1032, 2007.
© Springer-Verlag Berlin Heidelberg 2007

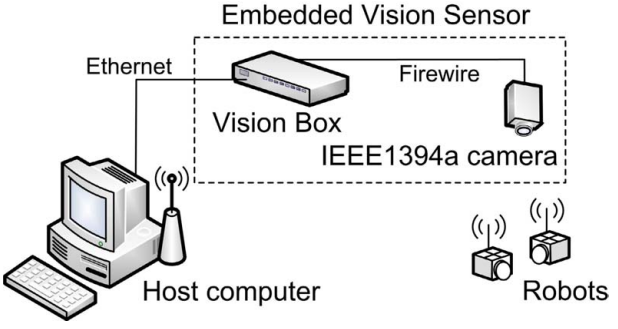


Fig. 1. Schematic overview of system configuration. The Embedded Vision Sensor consists of an Embedded Vision System (Vision Box) and a FireWire-camera.

(DSPs) or Field-programmable Gate Arrays (FPGAs). For the majority of non-autonomous systems, PC solutions are used, because development of algorithms is straightforward and state-of-the-art PCs offer a high computing power, too.

In [3] Andrzej Sluzek and Phung Khoi Duy Minh presented a vision module completely implemented on FPGAs. They developed this embedded vision module to be used as embedded sensor for robot soccer. A technology feature on *Developing and integrating FPGA coprocessors* was release by Paul Ekas and Brian Jentz [4]. The article demonstrates the use of an FPGA co-processor for a modulation device. They showed how to accelerate functions of a digital signal processor, by off-loading it to an FPGA. The computer vision and robotics group at the university of Girona, Spain, published an article proposing an architecture, which consists of a DSP and multiple FPGAs for real-time image processing [5]. They are using an array of FPGAs for distributed image calculation, controlled by a single DSP.

In summary, there is a lot of ongoing activity concerning this topic. The reasons can be found in the increasing computing power requirements for computer vision algorithms. In addition, FPGAs are getting cheaper and are therefore increasingly an alternative to DSPs. By combining the strengths of DSPs and FPGAs more advanced systems can be developed, offering a high flexibility for various applications.

3 System Concept

Figure 1 shows a schematic overview of the system configuration. A team in the MiroSot league consists of a camera, mounted on top of the playing field, a host computer, for image processing and controlling tasks, and three (Small League), five (Middle League) or eleven (Large League) cuboid robots limited to 7.5 cm in size. For our approach the complete image processing task was off-loaded to an embedded vision system, called *Vision Box* (see chapter 4), which is connected via ethernet to the host computer. The camera was connected to the embedded system via FireWire IEEE 1394a.

Therefore, the *Vision Box* and the camera act as an embedded vision sensor, which delivers the desired positions of the robots and the ball on the play field. By using such a system configuration the vision task is fully separated from the controlling task. This allows a modular development of the system components, and makes it easier to handle the overall system.

4 Realization of the Embedded Vision Sensor

4.1 Hardware Architecture

The hardware board used for the realization of the embedded vision system is a so called VisionBox MPS 2 from Strampe Systemelektronik [6]. The board consists of a TI TMS320C6414 fixed-point DSP [7] and an Altera EP1C3 FPGA [8]. The FPGA is connected to the DSP via the External Memory Interface B (EMIFB) [9], which has a bus width of 32 bits and a clock frequency of 66 MHz. The FPGA was originally intended to be used as an interface module between the DSP and external hardware. Here, the FPGA serves as a co-processor to boost the performance of some critical tasks executed on the embedded vision system. With 2900 logic elements, approx. 58Kbit RAM and 66 MHz clock rate, the FPGA used is the smallest available of the Altera Cyclone I family. Therefore, the integration of the co-processor into the FPGA was very challenging.

The color camera is a Basler A602fc with a resolution of 640 x 480 pixels and a maximum framerate of 100 fps. The output of the camera is a raw Bayer-filtered image. The lens is a Pentax C60607TH with a focal length of 6 mm. The camera is connected via FireWire IEEE 1394a to the VisionBox.

4.2 Vision Algorithm

The task of the algorithm is to analyze a camera image for determining the positions of the objects on the playing field. The vision algorithm for extracting robots and the ball starts with the conversion of the camera image from raw Bayer format to RGB (red, green, blue) color images. Based on the RGB-values of each pixel, the black background of the playing field is removed using a threshold method. The image is converted from RGB color space into HSV (hue, saturation, value) color space, for a robust, color based segmentation. Each pixel of the image is assigned to an appropriate color class, using a combination of a minimum distance and threshold technique [10]. Pixels with the same color class are merged to color regions and a region growing technique is applied. Finally, by analyzing the location of the color regions the position and viewing angle of the team robots are identified. After these steps, coordinates of the robots and the ball are available and are transmitted to the host using User Datagram Protocol (UDP) messages.

Figure 2(a) shows a part of a typical camera image, as it is received by the vision algorithm for image processing. The result of the computer vision algorithm is illustrated in Figure 2(b). The following information has to be extracted from the camera image:

- Coordinates (x,y) of the ball.
- Coordinates (x,y) and viewing angle α of all team robots.
- Coordinates (x,y) of all opposing robots.

For that task, the objects on the playing field have pre-defined color markings: The color of the ball has to be orange. The robots must be colored with a team color, which is either blue or yellow. Additionally the team robots can be marked with other colors to distinguish between the individual robots. The computer vision algorithm can be illustrated as a module, receiving the camera image as an input and delivering the coordinates as an output.



(a) (b)

Fig. 2. Camera image of the playing field (a) and the determined positions of the objects (b)

According to [11] a computer vision algorithm can be divided into three types of processes: low-, mid- and high-level processes. Low-level processes apply primitive operations on whole images. The input and output of a low-level process are images. Mid-level processes extract some features from an image, for example the positions of color regions. These features are interpreted by high-level processes. The flow chart of the robot soccer vision algorithm is shown in Fig. 3. The low-level processes of the robot soccer algorithm are:

- Bayer interpolation: Conversion of the camera images from raw Bayer format to RGB color images [12].
- Background filtering: The black background of the playing field is removed using a threshold method.
- HSV transformation: The image is converted from RGB color space into HSV color space for a robust, color based segmentation [13].
- Segmentation: The pixels of the image are assigned to an appropriate color class, using a combination of a minimum distance and threshold technique.

After passing these processes the algorithm continues with the mid-level process:

- Region detection: Pixels with the same color class are merged to color regions. The applied region growing technique is outlined in [14].

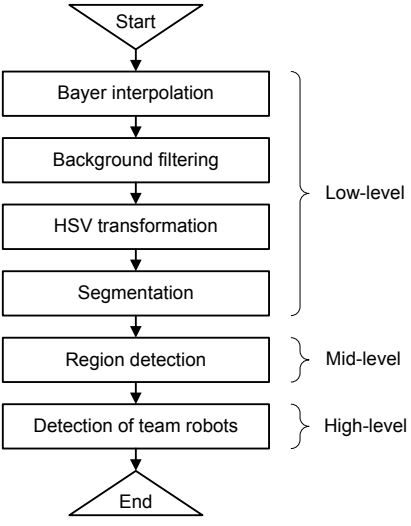


Fig. 3. Flow chart of the computer vision algorithm

The results of this process are passed on to the high-level process:

– Detection of team robots: By analyzing the location of the color regions the position and viewing angle of the team robots can be identified.

After these processes, the coordinates and orientations of the robots and the ball are extracted and can be sent to the host computer.

4.3 Performance Analysis of the Algorithm

To speed up the calculation of the algorithm on the embedded vision system, parts of the algorithm have to be selected, which are qualified for being implemented on FPGAs. The criteria for these selections are, that they can be

– accelerated, and
– implemented efficiently on parallel hardware.

The first criteria means that the part to be optimized has to be very time consuming on the DSP. No advantage can be made by optimizing parts of the algorithm, which need only a short duration of the overall calculation time. In this context the so called 80/20-rule of thumb is used: Commonly 80% of the calculation time is caused by 20% of the program code. To locate these 20% of code, a code profiling is carried out, measuring the runtime of all parts of the algorithm.

The second criterion addresses whether if the algorithm can be implemented efficiently on parallel hardware. Generally, every piece of code, running on a DSP, can be implemented on an FPGA, too. However, the efficiency of the

implementations varies for different types of codes. For example, parts repeating a simple operation in a loop can be implemented on FPGAs very efficiently. On the contrary, implementation of code with complex branches and varying operations may be very complicated on FPGAs.

Algorithm parts most suitable for an implementation on parallel hardware are usually found in the low-level processes, because they most likely fulfill the mentioned criteria. These processes apply simple operations on individual pixels or small pixel groups of an image, and can therefore be implemented on an FPGA efficiently.

The profiling of the algorithm revealed that nearly 60% of the processing time is consumed by the background filtering, HSV transformation and segmentation processes. Hence, these functions have been chosen for implementation on the FPGA co-processor. The hardware for exchanging data between the DSP and the FPGA was already available on the *VisionBox*. It offers a FIFO for sending and receiving data packages via the external memory interface with a size of 512 bytes. The co-processor was described using VHDL [15] and synthesized with Altera's Quartus II Design [16] Software. Table 1 shows a summary of the synthesis results. The EMIF FIFO unit and the implemented algorithm steps together needed 83 % of the total logic elements. About half of the available memory was used for this implementation.

Table 1. Synthesis results for the FPGA co-processor

Parameter	Logic elements	Memory bits
EMIF FIFO	482	16640
Algorithm	1925	14328
Total	2407	30968
Usage	83 %	52 %

4.4 Optimization Results

As described before, we optimized the implementation of the vision algorithm by offloading parts of it to a co-processor. Figure 4 presents the results of an analysis of three different approaches we studied. First, we did a standard implementation on the DSP, without any support of the FPGA co-processor. Secondly, we analyzed conventional co-processing, where the DSP sends data to the FPGA co-processor and gets the results back after the calculation. Finally, we tested an new approach, where the workload is optimally distributed between DSP and FPGA co-processor [17].

The original DSP version needed 16 ms for execution. When using a conventional co-processing technique an execution time of 18.7 was needed, which is even higher than the DSP version. The reasons for this can be found in the slow transfer rate between the DSP and the FPGA, and the low clock rate of the FPGA of the embedded system used. The co-processor routine receives an RGB image as an input and returns an 8-bit single channel image as an output. With

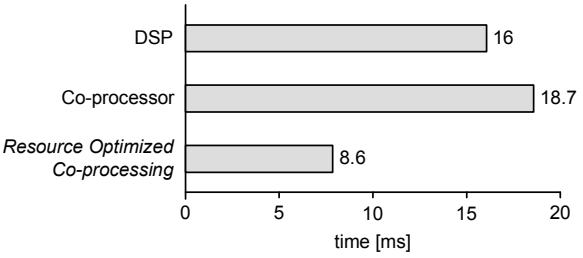


Fig. 4. Processing time for background filtering, HSV transformation, and segmentation. Results are shown for different co-processing methods

a size of 640x480 pixels, a total of 1,228,800 bytes have to be transferred. Using the following formula,

$$t_{Transfer} = \frac{NumOfBytes}{bw \cdot f_{bus}} \tag{1}$$

with $bw = 4\ bytes$, and a clock rate f_{bus} of 66 MHz of the EMIFB, the transfer time calculates to $t_{Transfer} = 4.7\ ms$. The implementation of the algorithm can process one byte per cycle. With the given clock rate of 66 MHz and following formula,

$$t_{Process,FPGA} = \frac{NumOfPixels\ \cdot CyclesPerPixel}{f_{FPGA}}. \tag{2}$$

the process time can be calculated as $t_{Process,FPGA} = 14\ ms$. By adding the transfer time and process time, the total execution time for the FPGA co-processor of 18.7 ms can be obtained.

By using our new approach of co-processing, *Resource Optimized Co-Processing,* the execution time was reduced to 8.6 ms, which means the routine was processed nearly twice as fast.

5 Conclusion and Outlook

In this contribution we presented an embedded vision sensor to be used for robot soccer. We achieved an embedded vision sensor system, which is able to work at full camera framerate (60fps) with full image resolution (640x480 pixels) without the need of any resources of the host computer.

The sensor is based on a DSP/FPGA co-processor system. The vision algorithms are specifically designed to optimally utilize the resources of the embedded vision system. Due to our approach the host computer is freed from the compute-intensive vision tasks. Therefore, a high-end computer is no longer needed.

Currently we are analyzing additional algorithms to identify potential new candidates for performance boosting. Candidates are forward collision warning and lane detection algorithms for autonomous driving of intelligent vehicles and autonomous robots.

References

[1] Mahlknecht, S., Oberhammer, R., Novak, G.: A Real-time Image Recognition System for Tiny Autonomous Mobile Robots. In: Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium, IEEE Computer Society Press, Los Alamitos (2004)

[2] FIRA: Federation of International Robotsoccer Association (April 2006), http://www.fira.net/

[3] Sluzek, A., Minh, P.K.D.: Embedded Vision Module for Robot-soccer, Nanyang Technological University, Singapore, SWPS, Warszawa (Poland) (2006)

[4] Ekas, P., Jentz, B.: Developing and Integrating FPGA Coprocessors. Embedded Computing Design Magazine (2003)

[5] Ballke, J., Marti, J., Ridao, P., Amat, J.: A New FPGA/DSP-Based Parallel Architecture for Real-Time Image Processing. Computer Vision and Robotics Group, Institute of Informatics and Applications, University of Girona (2004)

[6] Strampe Systemelektronik GmbH & Co. KG : Vision Box MPS2 (August 2006), http://www.strampe.de/

[7] Texas Instruments Incorporated: TMS320C64x Technical Overview Literature Number: SPRU395 (February 2000)

[8] Altera Cooperation: Cyclone FPGA Family Datasheet (2005), http://www.altera.com/literature/hb/cyc/cyc_c5v1_01.pdf

[9] Texas Instruments Incorporated: TMS320C672x DSP External Memory Interface (EMIF) User's Guide. Literature Number: SPRU711B (May 2006)

[10] Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. PWS Publishing (1999)

[11] Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Pearson Education International, London (2002)

[12] Brainard, D.: Bayesian method for reconstructing color images from trichromatic samples. In: Proceedings of the IS&T 47th Annual Meeting, Rochester, NY (1994)

[13] Ford, A., Roberts, A.: Colour Space Conversions (1975), http://www.poynton.com/PDFs/coloureq.pdf

[14] Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis and Machine Vision. PWS Publishing (1999)

[15] Ashenden, P.J.: The Designer's Guide To VHDL. Morgan Kaufmann, San Francisco (2002)

[16] Alter Coporation: Quartus II Design Software Product Description (June (2006), http://www.altera.com/products/software/products/quartus2/

[17] Rinnerthaler, F.: Optimization of Computer Vision Algorithms for Embedded Systems Using FPGA Co-processors. Diploma Thesis, University of Applied Sciences Hagenberg, Austria (June 2006)

MTVS: A Multi-task Active-Vision System

D. Hernandez, J. Cabrera, A. Naranjo, A. Dominguez, and J. Isern

IUSIANI, ULPGC, Spain,
dhernandez@iusiani.ulpgc.es*

Abstract. This paper describes the architecture of an active-vision system that has been conceived to ease the concurrent utilization of a visual sensor by several tasks on a mobile robot. We describe in detail the functional architecture of the system and provide several solutions to the problem of sharing the visual attention among multiple visual tasks. The system's design hides this complexity to client processes that can be designed as if they were exclusive users of the visual system. The results obtained in experiments with real mobile robots are also presented.

1 Introduction

The control of the gaze in an active vision system has attracted the attention of researchers under different approaches. Usually it is formulated as a problem of detection of significant points in the image. In this context, several aspects such as saliency, bottom-up vs. top-down control, computational modelling, etc, have been analyzed [1][2]. An alternative view considers the shared resource nature of the sensor, transforming the scenario into a management/coordination problem where the control of the gaze must be shared among a set of dynamic concurrent tasks.

From a more engineering point of view, researchers involved in the development of vision systems have been primarily concerned with the visual capabilities of the system in terms of performance, reliability, knowledge integration, etc. However, the majority of vision systems are still designed and integrated in a very primitive way according to modern software engineering principles, and there are few published results on how to control the visual attention of the system among several tasks that execute concurrently [3].

Posing a simple analogy to clarify these issues, when we execute programs that read/write files that are kept in the hard disk, we aren't normally aware of any contention problem and need not to care if any other process is accessing the same disk at the same time. This is managed by the underlying services, simplifying the writing of programs that can be more easily codified as if they have exclusive access to the device.

If we translate the former example to the context of vision systems, some important drawbacks can be easily identified.

* This work has been partially supported by Spanish Education Ministry and FEDER (project TIN2004-07087) and Canary Islands Government (project PI2003/160).

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1033–1040, 2007.
© Springer-Verlag Berlin Heidelberg 2007

- Vision systems tend to be monolithic developments. If several visual tasks need to execute concurrently this need to be anticipated since the design stage.
- It is very difficult to add new visual capabilities that may compete against other tasks for the attention of the system.
- As far as contention situations are dealt with internally and treated by means of ad-hoc solutions, the development of such systems does not produce any reusable technology for coping with these problems.

As a selection of related work, the following three systems can be mentioned. Dickmanns and colleagues [4] have studied the problem of gaze control in the context of their MarVEye Project, where an active multi-camera head is used to drive a car in a highway. In that project, several areas of interest are promoted and ranked by different modules of the control architecture using a measure of information gain.

Several groups have explored the problem of gaze arbitration in the humanoid robots scenario, both in simulation and with real robots. Seara et al. [5] [6] have experimented with a biped robot that used a combination of two tasks to visually avoid obstacles and localize itself. The decision of where to look next was solved in two stages. Firstly, each task selects its next preferred focus of attention as that providing the largest reduction of incertitude in the robot localization, or in the location of obstacles. In a second stage, a multiagent decision schema, along with a winner-selection society model, was used to finally decide which task was granted the control of gaze.

Sprague et al. [7][3] have designed a simulation environment where a biped robot must walk a lane while it picks up litter and avoids obstacles, using vision as the only sensor. These capabilities are implemented as visual behaviors using a reinforcement learning method for discovering the optimal gaze control policy for each task.

Similar in spirit to this related research, the motivation of our work is consequently two-fold: contribute to build a vision system more consistent from an engineering point of view, and to take a first step towards systems where the vision becomes integrated in an action context with higher semantic and cognitive level (an "intelligent" way of looking).

In the next sections we will present the proposed architecture, with its design objectives and main components. Some experimental results obtained on a real robot, along with conclusions and intended future development will be described in the last two sections.

2 MTVS Architecture

Motivated by the previously stated description of the problem we have designed and implemented MTVS (Multi-Tasking Vision System), a proposal of architecture for active-vision systems in multi-tasking environments. MTVS has been designed to deal with the scheduling of concurrent visual tasks in such a way that resource arbitration is hidden to the user.

More in detail, MTVS pursues the following objectives:

- The assignment of the gaze control to a task is based on a simple scheduler model, so that the behavior can be easily interpreted by an external observer.
- The client tasks are integrated in the system individually with no coordination requirements.
- The set of tasks managed (the activity) can change dynamically.
- The clients should not assume any a priori response time guarantee, though the system can offer high-priority attention modes.
- The system offers services based on a reduced set of visual primitives, in pre-categorical terms.

2.1 Internal Structure

The figure 1 shows an example with two clients and the basic elements making up the vision system architecture: a system server, a task scheduler and the data acquisition subsystem. Basically, the clients connect to the system server to ask for visual services with a given configuration (client A active). In response, the system server launches both a task-thread, to deal with internal scheduling issues, and a devoted server-thread that will be in charge of the interaction with the external client. The scheduler analyzes the tasks demands under a given scheduling policy and selects one to receive the gaze control. In combination, a second covert scheduler checks for compatibility between tasks to share images among them (FOA's overlapping). The data acquisition subsystem processes the different sensor data streams (images, head pose and robot pose) to generate as accurately as possible time stamps and pose labels for the served images.

2.2 Visual Services

Clients can connect to the vision system and use it through a number of pre-categorical low-level services. The MTVS services are built around basic visual capabilities or primitives that have also been explored by other authors [8]:

WATCH: Capture N images of a 3D point with a given camera configuration.
SCAN: Take N images while the head is moving along a trajectory.
SEARCH: Detect a model pre-categorically in a given image area.
TRACK: Track a model pre-categorically.
NOTIFY: Inform the client about movement, color or other changes.

Except for WATCH, the rest of primitives can be executed discontinuously, allowing for the implementation of interruptible visual tasks.

The clients also regulate their activity in the system by means of the messages they interchange with their devoted server. Currently, the following messages have been defined for a task: creation, suspension, reconfiguration (modify parameters, change priority, commute primitive on success) and annihilation.

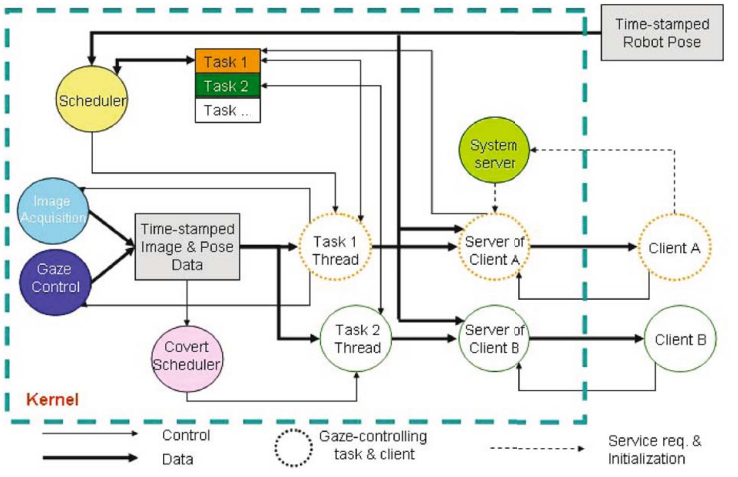


Fig. 1. Control Architecture: example with two clients

2.3 Scheduling Policies

Several scheduling policies have been implemented and studied inside MTVS. This analysis has considered two main groups of schedulers: time-based and urgency based schedulers.

Time-based schedulers

Three types of time-based schedulers have been studied: Round-Robin (RR), Earliest Deadline First (EDF) and EDF with priorities (EDFP). The prioritized RR algorithm revealed rapidly as useless in a dynamic and contextual action schema. First, it makes no sense to assign similar time slices to different tasks, and second, the time assigned used for saccadic movements, specially when a slow neck is involved, becomes wasted.

The EDF algorithm yielded a slightly better performance than RR, but was difficult to generalize as visual tasks are not suitable for being modelled as periodic tasks. The best results of this group were obtained by the EDFP algorithm combining critical tasks (strict deadline) with non-critical tasks. Each time a task is considered for execution and not selected its priority is incremented by a certain quantity [9].

Urgency-based schedulers

The concept of urgency is well correlated with a criteria of loss minimization, as a consequence of the task not receiving the control of the gaze within a time window. This measure can also be put into relation with uncertainty in many visual tasks.

Two schedulers have been studied in this group: lottery [7] and max-urgency. The lottery scheduler is based in a randomized scheme where the probability of a task being selected to obtain the gaze control is directly proportional to its

urgency. Every tasks has the possibility of gaining the control of the gaze, but the random unpredictability can sometimes produce undesirable effects.

The max-urgency scheduler substitutes the weighted voting by a direct selection of the task with higher urgency value. This scheme has produced acceptable results provided that the urgency of a task is reduced significantly after gaining the control of the gaze (similar to an inhibition of return mechanism).

3 Mobile Robot Experiments

A set of experiments were carried out to analyze the behavior of MTVS on a real robotic application. The basic experimental setup consists of two ActivMedia Pioneer robots, one with the basic configuration and the other mounting an active vision system formed by a Directed Perception PTU (neck) and a motorized Sony EVI-G21 camera (eye).

Two main tasks were combined along the different experiments: target following and obstacle avoidance. The target following task commands the active vision robot (pursuer) to detect and follow a special square target mounted on other robot (leader), trying to keep a predefined constant distance between them. The obstacle avoidance task looks for colored cylinders on the floor, estimating, as exactly as possible their 2D position. Kalman filtering is used to model both target and obstacles positions.

3.1 One-Task Experiments

As a reference for the maximum expected performance for each task some experiments where designed involving only one task.

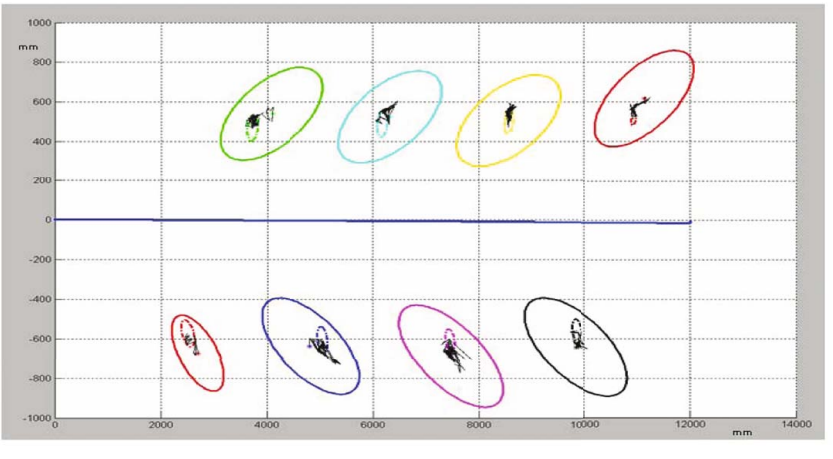


Fig. 2. Obstacle avoidance-Only experiment

Experiment 1: Obstacle avoidance only

The active vision robot is commanded to explore the environment looking for objects (yellow cylinders), trying to reduce their position uncertainty below a predefined threshold. The robot moves straight-line inside a corridor formed by 8 cylinders equally distributed in a zigzag pattern along the path. The figure 2 illustrates the robot path and the different detections for each localized object, including their first (larger) and minimum uncertainty ellipses. The results show how the robot was able to localize all the objects with minimum uncertainty ellipses ranging from 100 to 200 mm in diameter.

3.2 Multiple-Task Experiments

The multiple-task experiments consider an scenario in which each task computes its desired camera configuration and urgency and asks the MTVS scheduler to obtain the gaze control. The scheduler uses this information to select where to look next and how to distribute images. The obstacle avoidance task is extended to classify special configurations of objects as "doors" (two objects aligned perpendicularly to robot initial orientation with a pre-defined separation).

The urgency of the following task is computed as a function of the distance error, the robot velocity and the time. This urgency increases as the distance between the robots differs from the reference, the velocity is high and the elapsed time since the last image was received becomes larger.

The urgency of the obstacle avoidance task is computed separately for three possible focus of attention: front (the urgency increases when the robot moves towards visually unexplored areas), worst estimated object (the urgency increases as the position of a previously detected object is not confirmed with new images), and closest door (the urgency increases with narrow doors).

The first simple multiple-task experiment tries to illustrate the sharing images capability of MTVS. In experiment 3 a more complex scenario including doors is analyzed.

Experiment 2: Obstacle avoidance and robot following competing for the gaze (following priority)

In this experiment, the control of the gaze is only granted to the avoidance task when both the leader speed and the distance error are low. Typically, the following task performance is not affected significantly, but the avoidance task degrades yielding few objects localization with poor precision. As an example, the upper plot of the figure 3 presents the results of a non sharing run where only half the potential objects (all right sided due to the position of the closest obstacle) have been detected with large uncertainty ellipses. As the lower plot of the figure shows, the sharing of images permits a much better behavior of the obstacle avoidance task.

Experiment 3: Localize doors and robot following competing for the gaze (narrow and wide doors)

The configuration of objects used for this experiment consist of a set of four "doors": two narrow type (600 mm width) and two wide type (1500 mm width). All doors are located straight line in front of the robot, the first one (wide) three

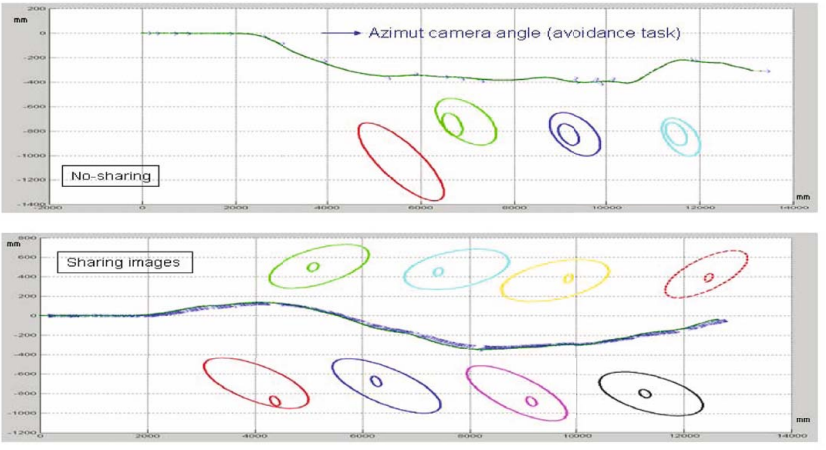


Fig. 3. Follow (priority) and avoidance experiment

meters ahead and the rest every 1.5 meters, alternating narrow and wide types. The leader robot is commanded to move at constant speed crossing the doors centered.

The figure 4 illustrates how the camera is pointed to both sides when crossing narrow doors. As a consequence of this behavior, the pursuer robot slows down when approaching a narrow door until the door extremes position have been estimated with the required precision (compare final error ellipses for narrow and wide doors). After traversing the door, the robot accelerates to recover the desired following distance from the leader.

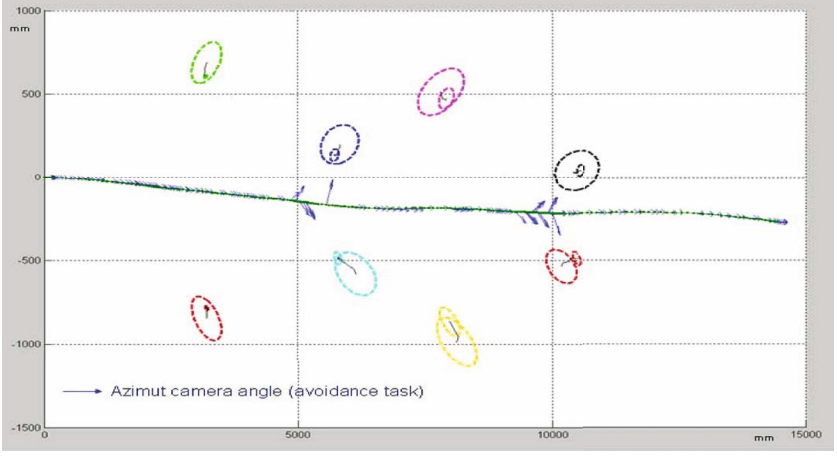


Fig. 4. Narrow and wide doors experiment

4 Conclusions and Future Work

In this paper we propose an open architecture for the integration of concurrent visual tasks. The clients requests are articulated on the basis of a reduced set of services or visual primitives. All the low level control/coordination aspects are hidden to the clients simplifying the programming and allowing for an open and dynamic composition of visual activity from much simpler visual capabilities.

Regarding the gaze control assignation problem, several schedulers have been implemented. The best results are obtained by a contextual scheme governed by urgencies, taking the interaction of the agent with its environment as organization principle instead of temporal frequencies. Usually, a correspondence between urgency and uncertainty about a relevant task element can be established.

The system described in this paper is just a prototype, mainly a proof of concept, and it can be improved in many aspects. The following are just a few of them. We plan to improve the adaptability of the system to different active vision heads (hardware abstraction). A first step is to consider the extension of the system to be applied over a binocular system, where new problems like eye coordination, vergence and accommodation must be tackled. Other issues include the need for an acceptance test on new service requests to avoid overloading the system, and introducing homeostasis to make perception more robust.

References

1. Arkin, R. (ed.): Behavior-Based Robotics. MIT Press, Cambridge (1998)
2. Itti, L.: Models of bottom-up attention and saliency. In: Itti, L., Rees, G., Tsotsos, J.K. (eds.) Neurobiology of Attention, Elsevier Academic Press, Amsterdam (2005)
3. Sprague, N., Ballard, D., Robinson, A.: Modeling attention with embodied visual behaviors. ACM Transactions on Applied Perception (2005)
4. Pellkoffer, M., Lützeler, M., Dickmanns, E.: Interaction of perception and gaze control in autonomous vehicles. In: SPIE: Intelligent Robots and Computer Vision XX: Algorithms, Techniques and Active Vision, Newton, USA (2001)
5. Seara, J.F., Lorch, O., Schmidt, G.: Gaze Control for Goal-Oriented Humanoid Walking. In: Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids), Tokio, Japan, pp. 187–195 (November 2001)
6. Seara, J.F., Strobl, K.H., Martin, E., Schmidt, G.: Task-oriented and Situation-dependent Gaze Control for Vision Guided Autonomous Walking. In: Proceedings of the IEEE/RAS International Conference on Humanoid Robots (Humanoids), Munich and Karlsruhe, Germany (October 2003)
7. Sprague, N., Ballard, D.: Eye movements for reward maximization. In: Advances in Neural Information Processing Systems, vol. 16, MIT-Press, Cambridge (2003)
8. Christensen, H., Granum, E.: Control of perception. In: Crowley, J., Christensen, H. (eds.) Vision as Process, Springer, Heidelberg (1995)
9. Kushleyeva, Y., Salvucci, D.D., Lee, F.J.: Deciding when to switch tasks in time-critical multitasking. Cognitive Systems Research (2005)

DES-Based Coordination of Space-Sharing Mobile Robots

Elzbieta Roszkowska

Institute of Computer Engineering, Control, and Robotics,
Wroclaw University of Technology, Poland
ekr@pwr.wroc.pl

Abstract. This paper presents a rigorous approach to the coordination of space-sharing mobile robots, based on the Discrete-Event-System (DES) formalism. The proposed solution provides a closed-loop, scalable control model, ensuring in a provable manner the correct concurrent robot movement. The control logics guarantee the occurrence of no collisions or deadlocks, and that each robot is able to complete its mission in a finite time. The event-driven character of the robot dynamics representation allows the implementation of the results as both a central and as a distributed supervisory control system.

Keywords: Multiple mobile robot system, collision and deadlock avoidance, supervisory control, DES-based model, event-driven control.

1 Introduction

The research on the operation of multiple mobile robots moving in a common space have mostly concentrated so far on motion planning with respect to collision avoidance and performance optimization [1]. From the viewpoint of the control of a multi-robot system, the motion planning approach has two basic shortcomings. In the first place, at the coordination level, the realization of motion plans is an open-loop control policy, based on deterministic time functions. Such a control is very sensitive to the system randomness, which, given the autonomous and asynchronous operation of the robots, makes the eventual applicability of these open-loop control plans highly questionable. Secondly, even though physical collisions can be avoided when using time-based dynamic system models, they give no insight into another critical control problem, that of deadlock-handling.

An alternative research direction and an attempt to remedy the above mentioned deficiencies are control methods that view a multiple mobile robot system (MMRS) as a group of asynchronously operating independent agents. The coordination of these agents has the form of a supervisory control that can be realized either centrally or in a distributed way, and involves decisions calculated in real-time, based on the feedback from the system. While most of the papers in this area exclusively focus on, and propose various procedures for collision avoidance, e.g., [2], a number of authors also recognize the importance

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1041–1048, 2007.
© Springer-Verlag Berlin Heidelberg 2007

of the deadlock handling problem. The latter contributions deal with deadlock avoidance in a specific environment, such as narrow corridors [3,4], or solve the problem through specific design of intersections points, equipped with buffering areas [5]. A more general approach model of a mobile robot system is assumed in [6], where deadlocks are not prevented, but allowed to occur, then detected and, if possible, resolved.

In contrast to the above, software oriented rather than formal mathematical models, this paper presents a rigorous approach to the coordination of space-sharing mobile robots, based on the Discrete-Event-System (DES) formalism [7]. The proposed solution provides a closed-loop, scalable control mechanism, ensuring in a provable manner the correct concurrent robot movement. The control logic guarantees the occurrence of no collisions or deadlocks, and that each robot is able to complete its mission in a finite time. This paper follows the general scheme of the supervisory control introduced in our earlier works [8,9], but conceives a broader control problem, including event-driven, high-level control of the robot dynamics, and the coordination of such robots in the case when they can dynamically change their paths.

2 DES Model of Robot Motion Processes

We consider a MMRS (Multiple Mobile Robot System) consisting of a set of independent, autonomous robots A_i, $i \in 1, \ldots, n$, that operate in a common 2D workspace $W \subset \mathcal{R}^2$. The robots can communicate among themselves, e.g., through the distributed blackboard mechanisms [10], or through peer to peer messaging and/or message broadcasting [11]. Each robot has a trajectory planner, which plans its own trajectory from the initial to the final location. At this stage no positional constraints introduced by any other robot are taken into account. In order to avoid collisions, the robots refine their paths and/or velocity profiles based on the following high-level control scheme.

The path p_i of each robot A_i is partitioned into a finite number of *sectors* $p_i^1, p_i^2 \ldots, p_i^{n_i}$. The motion of a robot A_i is guarded by a supervisor (discussed in more detail in the next section), which decides for the robot whether a transition to its new sector is admissible in the running state. If this is not the case, the supervisor first tries to modify the path of A_i, and if it turns out impossible, then its velocity profile. The former modification aims at substituting the unavailable path sector p_i^{k+1} with a bypass bp_i^{k+1} to sector p_i^{k+2} that is not impeding the progress of the other robots. The latter modification assumes that the robot remains on its path, but slows down and, possibly, comes to a stop, waiting until the situation changes and it can safely resume further travel.

In the proposed model, we do not make any assumptions on the robots' dynamics except of that the shortest time to travel a sector is sufficient to communicate with the supervisor, let it make decision on the next motion stage (whether or not, and if yes then how, to modify the trajectory), and (if necessary) to stop within the sector. This implies, in particular, that the length of a sector must be greater than the stop-distance *sd*, i.e., the distance that is always sufficient

and acceptable for a robot to come to a full stop. Then, the potential behavior of a robot can be represented by an event driven system, a deterministic finite state automaton (DFSA) [7], defined as follows.

Definition 1. *Let $P_i = \{p_i^k : k = 1, \ldots, n_i\}$ be the path sector sequence of robot A_i. The automaton model of A_i is a DFSA $Rob_i = (S_i, E_i, \Gamma_i, \delta_i, s_{0,i}, S_{M,i})$ s.t. (such that):*

1. *S is the set of states constituted by the pairs $s = (Q, \mu)$, where:*
   - *Q is the set of sector numbers, to which robot A_i has been granted access, that can take the following forms: i) $Q = \{k\}$, $k \in 1, \ldots, n_i\}$, or ii) $Q = \{k, k+1\}$, $k \in 1, \ldots, n_i - 1$.*
   - *$\mu \in \{0, \uparrow, \rightarrow, \downarrow, \sim\}$ is the operation-mode symbol denoting, respectively, readiness to start the travel process, acceleration, stable motion (with the velocity $v(p_i^k)$ determined by the initial trajectory plan for robot A), deceleration (with retardation sufficient to come to rest within the safe distance), and no-motion.*
2. *$E = \{g,\ b,\ f,\ d,\ start,\ max,\ stop\}$ is the set of events, that represent, respectively, grant to access next sector, grant to access a new-calculated bypass, coming at the critical distance, the 'stop-distance' sd to the end of the sector, completion of the transition to a new sector, start of the travel process, achievement of the desired stable motion (end of acceleration), and achievement of a full stop.*
3. *$\Gamma : S \rightarrow 2^E \subseteq E$ is the feasible event function s.t. $e \in \Gamma(s)$, i.e., event $e \in E$ is feasible in state $s = (Q, \mu)$ iff one of the following conditions holds:*
   - *$e = start$ and $\mu = 0$*
   - *$e = g$ or $e = b$ and $Q = \{k\}$ and $k < n_i$*
   - *$e = f$ and $|Q| = 1$ and $\mu = \rightarrow$*
   - *$e = d$ and $|Q| = 2$ and $\mu = \rightarrow$*
   - *$e = max$ and $\mu = \uparrow$*
   - *$e = stop$ and $\mu = \downarrow$*
4. *$\delta : S \times E \rightarrow S$ is the transition function defined for the pairs (s, e) s.t. event e is feasible in state $s = (Q, \mu)$, and given by*

$$s' = \begin{cases} (1, \uparrow) & \text{if} \quad \mu = 0 \\ (\{k, k+1\}, \uparrow) & \text{if} \quad e = g \ \wedge \ Q = \{k\} \ \wedge \ \mu \neq \rightarrow \\ (\{k, k+1\}, \rightarrow) & \text{if} \quad e = g \ \wedge \ Q = \{k\} \ \wedge \mu = \rightarrow \\ (\{k+1\}, \mu) & \text{if} \quad e = d \ \wedge \ Q = \{k, k+1\} \\ (Q, \downarrow) & \text{if} \quad e = f \\ (Q, \rightarrow) & \text{if} \quad e = max \\ (Q, \sim) & \text{if} \quad e = stop \end{cases}$$

5. *$s_0 = (\{1\}, 0)$ is the initial state*
6. *S_M, the set of marked states of Rob_i, is the singleton $S_M = \{(\{n_i\}, \sim)\}$, that is, the automaton only marks complete event sequences, that drive robot A_i to its goal.* ◇

Fig. 1 depicts the transition graph of the DFSA model Rob_i of an example robot A_i. Notice that the trajectory from the initial state, $(\{1\}, 0)$, to the final state, $(\{4\}, \sim)$, that includes the top row of the states corresponds to the initial robot's plan for traveling along path π, as no deceleration is enforced.

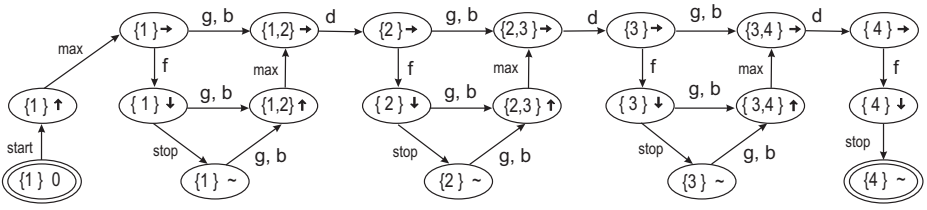


Fig. 1. Transition graph of the DFSA Rob_i for a robot A_i with 4-sector path

3 Collision Avoidance in MMRS

As mentioned in the previous section, the path p_i of each robot A_i is partitioned into a finite number of sectors. When traveling along sector p_i^k, robot A_i sweeps the space that will be called a *trace* and denoted by w_i^k. Based on the geometry of the traces, we can establish the conflict relation between path sectors.

Definition 2. *Consider an MMRS with n robots A_i, and let $P = \{p_i^k : i = 1, \ldots, n,\ k = 1, \ldots, n_i\}$ be the set of the robots' path sectors. Then, sectors $p_i^k \in P$ and $p_j^l \in P$ are in conflict iff*

1. *$i \neq j$, i.e., p_i^k and p_j^l are path sectors of distinct robots, and*
2. *$w_i^k \cap w_j^l \neq \emptyset$, i.e., the corresponding traces intersect.* ◇

Then, the collision avoidance can be ensured by the following policy:

Policy 1 *For each state of the system, for each pair of sectors that are in conflict, only one of them can host a robot at a time.* ◇

In order to enforce such a system behavior, the changing of sectors must observe a certain protocol. The transition of robot A_i from sector p_i^k to p_i^{k+1} requires that none of the robots has been granted access to a sector that is in conflict with p_i^{k+1}. If this is the case, then A_i obtains access to p_i^{k+1}, and thus, during the transition between the sectors, both p_i^k and p_i^{k+1} are exclusively accessible for A_i. However, as soon as the robot has left sector p_i^k, the rights of A_i to access this sector are withdrawn. In the opposite case, that is if any of the sectors that are in conflict with p_i^{k+1} has been acquired by another robot, the supervisor first tries to modify the path of A_i, and if this is impossible, then its velocity profile. The former modification aims at substituting the unavailable path sector p_i^{k+1} with a bypass, bp_i^{k+1}, to sector p_i^{k+2} that is not in conflict with any currently occupied sector p_j^l. The latter modification requires that the

robot slow down and, possibly, stop its motion until the situation changes and it can safely resume further travel. A supervisor that enforces the above described dynamics is constructed below.

Definition 3. *Consider a MMRS that consists of n robots given by their path sectors $P = \{p_i^k : i = 1, \ldots, n,\ k = 1, \ldots, n_i\}$, and assume that the first sector p_i^1 and the last sector $p_i^{n_i}$ of each robot A_i is not in conflict with any other sector $p_i^k \in P$. Let $Rob_i = (S_i, E_i, \Gamma_i, \delta_i, s_{0,i}, S_{M,i})$ be the automaton of robot A_i. Next, let $\mathcal{P}$ be any set of path sectors defined for the workspace of MMRS, and let $Con(p, \mathcal{P})$ be the set of all sectors $p \in \mathcal{P}$ that are in conflict with p. Moreover, let $\beta_i^k = \beta(p_i^k, s)$ be a function that for a MMRS state $s \in S = S_1 \times S_2 \ldots \times S_n$ and a sector p_i^k, $i = 1, \ldots, n$, $k = 1, \ldots, n_i$, either returns a bypass $\beta_i^k \in \mathcal{P}$ or the value* `null`*. Then, supervisor $Sp1 = (X, E, \Gamma, \delta, x_0, X_M)$ is a DFSA such that:*

1. *The state set X is constituted by the pairs $x = (q, \Pi)$ such that:*
   - *$q = (Q_i = (s_i; 1) : s_i \in S_i,\ i = 1, \ldots, n)$ is a vector of the first components of the robot's states, i.e., the numbers of the sectors currently accessible for the respective robots*
   - *$\Pi = \{\pi_i^k : i \leq n,\ k \leq n_i\} \subseteq \mathcal{P}$ is the set of remaining (i.e., not yet traveled) path sectors*
2. *The event set is given by $E = \bigcup_{i=1,\ldots,n} \{g, b, d \in E_i\}$.*
3. *$e \in \Gamma(x) \cap E_i$, i.e., event $e \in E_i$, $i = 1, \ldots, n$, is feasible in state $x \in X$ iff one of the following conditions holds:*
   - *$e = d$, i.e., return of the sector-access rights is always feasible,*
   - *$e = g \Rightarrow Q_i = \{k\} \ \wedge\ k < n_i \ \wedge\ Con(\pi_i^{k+1}), \Pi) \cap \bigcup_{j=1,\ldots,n} Q_j = \emptyset$, i.e., granting access to a path sector is feasible iff this sector is not in conflict with any of the currently occupied sectors*
   - *$e = b \Rightarrow g \notin \Gamma(x) \wedge Q_i = \{k\} \wedge k < n \wedge \beta_i^{k+1} \neq null \wedge Con(\beta_i^{k+1}, \Pi \setminus \pi_i^{k+1}) \cap \bigcup_{j=1,\ldots,n} Q_j = \emptyset$, i.e., a bypass for sector π_i^{k+1} is only calculated if π_i^{k+1} is not accessible, and if a non-conflict bypass $\beta_i^{k+1} = \beta(p_i^{k+1}, s)$ is found then A_i can be granted access to it.*
4. *For each event $e \in \Gamma(x) \cap E_i$, the next state $x' = \delta(x, e)$ is a vector $x' = (Q_1', Q_2', \ldots, Q_n', \Pi')$ such that:*
   - *$\forall j \neq i,\ Q_j' = Q_j$, and $Q_i' = (\delta_i(s_i, e); 1)$, where $s_i = (Q_i, \rightarrow)$*
   - *if $e = g$ then $\Pi' = \Pi$*
   - *if $e = d$ then $\Pi' = \Pi \setminus \{\pi_i^k\}$, where $k \in Q_i = \{k, k+1\}$*
   - *if $e = b$ then Π' is obtained by the substitution $\pi_i^{k+1} := \beta_i^{k+1}$*
5. *The initial state $x_0 = (q_0, P)$, where $q_0 = (Q_{0,i} = (s_{0,i}; 1) : i = 1, \ldots, n)$.*
6. *The set of marked states is the singleton $X_M = \{x_M\} = \{(q_n, \Pi_n)\}$, where $q_n = (\{n_1\}, \{n_2\}, \ldots, \{n_n\})$, and $\Pi_n = \{p_i^n \in P\}$.* ◇

In a supervised MMRS, event $e \in E_i \cap E$, $i \in 1, \ldots, n$, can only occur if it is feasible in both automaton Rob_i and $Sp1$. This implies the following theorem.

Theorem 1. *A MMRS supervised by Sp1 is collision free.*

Proof. When guarded by Sp1, a robot can only execute event g or b (that result in its transition to the next path sector) in a state x such that $g \in \Gamma(x)$ or $b \in \Gamma(x)$, respectively. Since the logics of Γ implement Policy 1, a collision among robots cannot occur. ◇

4 Deadlock Avoidance in MMRS

The incorporation of the supervisor Sp1 into the the MMRS prevents robot collisions, yet does not ensure, in general, that all the robots can complete their missions, i.e., that the marked state x_M can be reached. A phenomenon that makes it impossible is the deadlock. A simple example of a situation that can lead to a deadlock is presented in Fig. 2. Part (a) shows a configuration where a deadlock can still be prevented by forbidding the bottom robot enter its next sector. Part (b) depicts a state where any further motion of the robots unavoidably leads to a deadlock, and part (c), the deadlock itself.

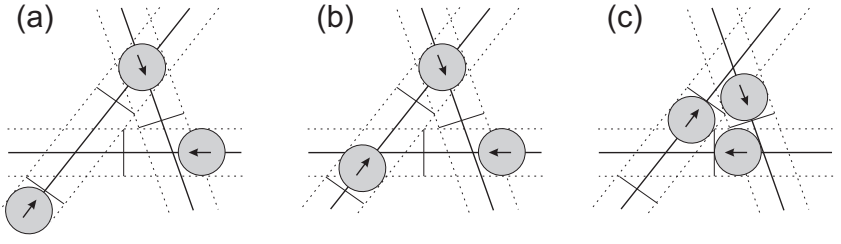


Fig. 2. Robot configuration that can lead to a deadlock

In order to make such incidents impossible to happen, it is necessary to impose more restrictive constraints on the event occurrence than those provided by SPR1. More specifically, we need to substitute the feasible-event function $\Gamma(x)$ with function $\Gamma^*(x) \subseteq \Gamma(x)$, $x \in X$, that lessens the domain[1] of the transition function $\delta(x, \sigma)$, thus producing a more restrictive function $\delta^*(x, \sigma)$, which ensures that the behavior of MMRS enjoys the following property:

Property 1. For each string $\sigma \in E^*$ s.t. function $x = \delta^*(x_0, \sigma)$ is defined, there exist string $\sigma' \in E^*$ s.t. function $\delta^*(x, \sigma')$ is defined and $\delta^*(x, \sigma') = x_M$. ◇

We enforce this property using the concept of a λ-*ordered state*.

Definition 4. *Let* $x = (q, \Pi) = (Q_1, Q_2, \ldots, Q_n, \Pi)$ *be a state in supervisor Sp*1*, and let* $Priv(i) = \{x : p_i^x \in \Pi \setminus \bigcup_{k=1}^{n_i} Con(p_i^k, \Pi)\}$*. Then, state* x *will be*

[1] In order to capture state transitions resulting from strings of events $\sigma \in E^*$, we assume that function $\delta(x, e)$ is extended to $\delta(x, \sigma)$, i.e., from the domain $S \times E$ to the domain $S \times E^*$, in the following manner: (i) $\forall x \in X,\ \delta(x, \epsilon) = x$, and (ii) $\forall x \in X, \forall \sigma \in E^*, \forall e \in E,\ \delta(x, \sigma e) = \delta(\delta(x, \sigma), e)$, where ϵ denotes the *empty string*.

called λ-ordered if there exists a bijective map $\lambda : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ s.t. for each pair $i, j \in \{1, \ldots, n\}$ the following property holds:

$$\lambda(i) < \lambda(j) \Rightarrow \forall l,\ l' < l < l'',\ Con(\pi_i^l, \Pi) \cap Q_j = \emptyset$$

where: $l' = max(k \in Q_{\lambda(i)})$, and $l'' = min(k \in Priv(\lambda(i)) \cup \{n_{\lambda(i)}\})$ $\diamond$

In words, $Priv(i)$ is the set of the numbers of the remaining sectors of robot A_i (i.e., those that robot A_i has still to travel) that are not in conflict with any remaining sector of any other robot. l' indicates the number of the current path sector of A_i, and l'', the number of the nearest non-conflict sector, $p_i^{l''} \in Priv(i)$. Moreover, in a λ-ordered state, if $\lambda(i) < \lambda(j)$ then no sector p_i^l, $l' < l < l''$, is in conflict with any sector $p_j^k \in Q_j$.

Definition 5. *For a MMRS guarded by supervisor $Sp1 = (X, E, \Gamma, \delta, x_0, X_M)$, supervisor $Sp2 = (X, E, \Gamma^*, \delta^*, x_0, X_M)$ is a DFSA such that:*

- $e \in \Gamma^*(x) \Leftrightarrow e \in \Gamma(x)$ *and state x is λ-ordered*
- $\delta^*(x, e) = x' \Rightarrow \delta(x, e) = x'$ $\diamond$

Theorem 2. *A MMRS guarded by Sp2 is collision and deadlock free.*

Proof. Since Sp2 is a restriction of Sp1, then by Theorem 1, MMRS guarded by Sp2 is collision free. To demonstrate that it is also deadlock free, it is sufficient to show that Sp2 enjoys Property 1. To see that, notice that in a λ-ordered state x robot A_i s.t. $\lambda(i) = 1$ can transition from sector $p_i^{l'}$ to sector $p_i^{l''}$ as no path sector p_i^l, $l' < l''$, is in conflict with any sector that any other robot has been granted access to. Next, since $p_i^{l''} \in Priv(i)$ then the same holds for robot A_i s.t. $\lambda(i) = 2$, and then all the remaining robots A_i can attain their sectors in $Priv(i)$, in the order of $\lambda(i) = 3, \ldots, n$. Since these sectors are not in conflict with any of the remaining sector of any robot, now the robots can attain their respective last sectors, one by one, in any order. This brings the system to the marked state x_M. $\diamond$

5 Conclusions

This paper extends our earlier contribution [8,9] to the coordination of mobile robots, by developing a supervisory control model for a new class of MMRS. In particular, a specific assumption for this class of a system is that the robots independently plan their trajectories, and are allowed to modify dynamically, during their travel, both their paths and velocity profiles. The coordination concept is based on a two-level, event-driven DFSA model of the robots' dynamics, discretized through partitioning the path of each robot into a number of sectors. The movement of a robot within a path sector is controlled independently from the movement of the other robots, while sector crossing is guarded by a policy, which, for calculating the control, requires the knowledge of the state of all the robots. Both these mechanisms have the form of DES models - robot automaton *Rob*, and supervisor automata $Sp1$ and $Sp2$, and their correctness with respect

to such a behavioral requirement as enforcement of no collision or deadlocks have been formally proved.

As this work is mainly focused on the robots' coordination logic, further research is needed towards implementation of the concept. It should be emphasized, in particular, that from the viewpoint of the correct system behavior, the path partition into sectors can be arbitrary. However, it can influence MMRS' performance, and thus the selection of the best partition can be considered as a separate problem, posed in the context of efficient workspace sharing. Another problem is the development of methods to compute the conflict relation among sectors, which can either be based on symbolic or numeric approach.

Moreover, while a copy of the robot automaton should be implemented in each robot, the supervisor can be realized in the whole spectrum of ways - as a central system, controlling the robots in a feedback loop, as a distributed system with a copy of $Sp2$ in each robot and a communication system to broadcast the local states, and as a partially centralized system, with multiple copies of the supervisor, but central memory accessible for all the robots.

References

1. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
2. Bruce, J., Veloso, M.: Safe multirobot navigation within dynamics constraints. Proceedings of the IEEE 94(7), 1398–1411 (2006)
3. Lee, Y., Gupta, K., Payandeh, S.: Motion planning of multiple agents in virtual environments using coordination graphs. In: ICRA 2005, pp. 378–383. IEEE Computer Society Press, Los Alamitos (2005)
4. Lee, Y., Gupta, K.: Motion planning of multiple agents in virtual environments on parallel architectures. In: ICRA 2007, pp. 1009–1014. IEEE Computer Society Press, Los Alamitos (2007)
5. Wang, J., Premvuti, S.: Distributed traffic regulation and control for multiple autonomous mobile robots operating in discrete space. In: ICRA 1995. IEEE Int. Conf. Robotics and Automation, vol. 2, pp. 1619–1624 (1995)
6. Jager, M., Nebel, B.: Decentralized collision avoidance, deadlock detection, and deadlock resolution for multiple mobile robots. In: IROS 2001. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, vol. 3, pp. 1213–1219 (2001)
7. Cassandras, C., Lafortune, S.: Introduction to Discrete Event Systems. Kluwer Academic Publishers, Boston, Dordrecht, London (1999)
8. Roszkowska, E.: Supervisory control for multiple mobile robots in 2D space. In: RoMoCo 2002. Int. Workshop on Robot Motion Control, pp. 187–192. Publishing House of Poznan University of Technology, Poznan (2002)
9. Roszkowska, E.: Provably correct closed-loop control for multiple mobile robot systems. In: ICRA 2005, pp. 2810–2815. IEEE Computer Society Press, Los Alamitos (2005)
10. Harmon, S., Aviles, W., Gage, D.: A technique for coordinating autonomous robots. In: ICRA 1986. IEEE Int. Conf. Robotics and Automation, pp. 2029–2034. IEEE Computer Society Press, Los Alamitos (1986)
11. Wang, Z., Zhou, M., Ansari, N.: Ad-hoc robot wireless communication. In: IEEE Int. Conf. System, Man, and Cybernetics, vol. 4, pp. 4045–4050 (2003)

CASIMIRO, The Sociable Robot

O. Déniz, M. Castrillón, J. Lorenzo, and M. Hernández

Departamento de Informática y Sistemas
Universidad de Las Palmas de Gran Canaria
Edificio de Informática. Campus de Tafira
35017 Las Palmas - Spain
odeniz@dis.ulpgc.es

Abstract. Sociable robots are gaining popularity among robotics researchers and cognitive scientists. These robots generally show abilities that include expressive power (face, voice, ...), locating, paying attention to and addressing people, etc. Such abilities fall into what is known as social intelligence in humans. The reproduction of social intelligence, as opposed to other types of human abilities, may lead to fragile performance, in the sense of having rather different performances between tested cases and future (untested) cases and situations. This limitation stems from the fact that our social abilities are mainly unconscious to us. This is in contrast with other human abilities that we carry out using conscious effort, and for which we can easily conceive algorithms and representations. The fragile performance mentioned above is nothing but overfitting. Thus, we propose to approach the problem using strategies followed in Machine Learning for avoiding overfitting. Specifically, complexity penalization and incremental design are translated to the broader ambit of robot design and development. The robot CASIMIRO is currently being developed following that approach.

Keywords: Social Robots, Human-Robot Interaction.

1 Introduction

In the last years there has been an upsurge of interest in sociable robotics. Starting from Kismet [2], many such robots have been built and are still being actively researched. All these robots share a common objective: interactivity. That is, they must engage in interactions with people. Of course, these interaction sessions should be as natural as possible. Thus, the robots should have basic abilities to detect people, address them, talk, express its emotions or intentions, etc.

Why so much interest in these abilities? Researchers have realized that these social abilities are in fact very important. Admittedly, we have always been building robots to emulate human intelligence and capacities. For many decades human intelligence was reduced to just logical-mathematical capacities. Tests were devised to measure such abilities and produce a single number for measuring intelligence. Notwithstanding, we now know that human intelligence is actually a set of abilities that encompasses many unrelated capacities.

This suggests that social intelligence aspects must be addressed if we are to build robots that imitate human intelligence. What is more, social intelligence could be even

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1049–1056, 2007.
© Springer-Verlag Berlin Heidelberg 2007

more important than other capacities. There is evidence that in primates social intelligence is one important prerrequisite for the evolution of non-social, domain-independent intelligence [6]. As an example, the highly elaborated ability of symbolization is suggested to be a social act of agreeing [3]. Some authors contend that social intelligence is also necessary for the development of generic intelligence in humans, see [12]. This 'Social Situatedness Hypothesis' (also, Machiavellian Intelligence Hypothesis) emphasizes the importance of designing and building robots that can interact with other agents.

This paper is organized as follows. Sections 2 and 3 describe the concepts of unconscious processing and *niche*. Based on these ideas, in Section 4 we give a set of recipes for developing this kind of robots. Section 5 briefly describes CASIMIRO, a prototype social robot. Finally, the main conclusions are outlined.

2 Unconscious Processes

When we deal with social intelligence, however, a problem appears. Most social abilities are unconscious. Some details remain hidden to our awareness, for it seems that, in healthy adults, an enormous amount of processing is done unconsciously.

Nowadays, the existence of unconscious processes in our brain seems to be beyond doubt. Freud's work already acknowledged that unconscious ideas and processes are critical in explaining the behavior of people in all circumstances. Helmholtz [10], studying vision, pointed out that even basic aspects of perception require deep processing by the nervous system. He argued that the brain constructs perceptions by a process of unconscious inference, reasoning without awareness.

Face recognition, an ability which obviously falls within the ambit of social intelligence, is another example of unconscious processing. We do not know what features in a face tell us that the face belongs to individual X. Yet, we carry out that process fast and robustly every day. Prosopagnosia is another rare neurological disorder characterized by a specific inability to recognize faces [7]. As is the case in blindsight, patients with prosopagnosia show an ability to perceive and process the visual input of which they claim not to be aware. They see faces, though they can not recognize them. These patients are further evidence for the existence of unconscious processing.

Many other examples of conscious/unconscious dissociation have been encountered. Researchers have demonstrated that this dissociation is present in perception, artificial grammar learning, sequence learning, etc., see [5] for descriptions of experiments. These findings suggest that the presence of unconscious influences on behavior is pervasive.

Why are some mental processes unconscious? Unconscious mental processes are fast, allowing us to do things like riding a bicycle without having to think about how to control each of our movements. Some authors contend that it is practice and habituation what makes details go unconscious [1]. Only novel, informative inputs trigger conscious processing (or "mental attention", which is a scarce resource). When events are easily predictable, they are no longer conscious [14]. As we habituate we have less and less conscious consideration for the stimuli (i.e. we put less mental attention to those redundancies). Being consciousness a scarce -limited- resource, this process would perfectly account for the progress of the individual that learning implies.

If practice and habituation is what makes details go unconscious, then social abilities should be relatively more unconscious, as they appear first in life and are always present in the acquisition of other abilities. In fact, unconscious processes are behind all or part of what we call social abilities, like face and language recognition (as opposed to other mental processes like for example solving a differential equation, which require conscious effort). ?

At this point it seems clear that there must be a difference in the process involved in building precision manipulators or calculus machines and that involved in building a robot that can socialize with humans as we do. In particular, if we do not know much of the algorithms and representations involved in our social abilities, how will our implementations be?

For some types of machines like robotic manipulators, one can extract a set of equations (or algorithms, representations,...) that are known to be valid for solving the task. Such equations would have been obtained after analytical effort, mainly related to kinematics. Once that these equations are stored in the control computer the manipulator will always move to desired points and therefore there is a sort of deductive process involved.

On the contrary, for social tasks, the robot design and development process is more of an inductive process (induction is a process by which, on the basis of the observed performance for a set of test cases, one aims at obtaining good results for unseen cases). In inductive processes one cannot have the guarantee of good performance for untested cases. This problem is more accentuated as less knowledge of the causal relation involved is available. In the case of reproducing social abilities, we have seen that there is little knowledge available to us, and thus we come to the conclusion that there is a serious risk of overfitting (i.e. having good performance for tested cases but poor performance of future -unseen- cases).

3 Niches

Even though we are trying to emulate the human ability, the robot will always perform in a much restricted environment. In ecology there is an appropriate term for that and we will borrow it here: *niche*. The niche is the range of environmental conditions (physical and biological) under which an organism can exist. For a robot, the niche would include aspects like room sizes, illumination, wall (background) colours, batteries, expected human presence, etc. That is, it is the range of conditions under which the robot is expected to work and perform well.

Ecology distinguishes two types of niche. The *fundamental* niche is the total range of environmental conditions that are suitable for existence. The *realized* niche is the range of conditions in which an organism actually performs at a given moment. The realized niche is generally narrower than the fundamental niche, mainly because of the effects of competition and predation from other species, see [11].

Humans themselves have their own fundamental and realized niches. For a given social task, for example, humans have a fundamental niche (the range of conditions under which the task is performed well). Obviously, if a human performs well in its fundamental niche then it also performs well in any realized niche inside. What we are

trying to achieve in our context is to reproduce the task in the same fundamental niche. We would like, for example, to achieve face detection under the same range of conditions that humans experience (i.e. variations in illumination, distances to people, etc.). The robot, however, is always expected to work in a much more restricted environment, its realized niche. Some particular cases that we ourselves encounter in our daily lives will never appear in the niche the robot is to work in. Obviously, throughout the robot development process we must be interested in minimizing error in its realized niche.

If we try to reproduce the ability as in the fundamental niche, with simple implementations we may obtain improvements in the realized niche. However, due to our limited knowledge, if we go too far and use too much of the available knowledge or intuitions we could be actually worsening performance in the realized niche. That, in fact, is something relatively frequent. Consider the problem of face detection. If we reflect on how we detect faces, we would say that by detecting the presence of certain features simultaneously, such as skin colour, eyes, mouth, etc. Sophisticated detection systems exist that use these and other features. However, it is our experience that only skin colour blobs with certain width-height ratios are sufficient to detect faces in many practical cases. Using a more elaborated or intuitive algorithm without extensive testing and tuning *in situ* usually leads to many typical faces going undetected (i.e. false negatives).

Note that the search for implementations that work well for the robot's realized niche is not an easy one. We still have the same lack of conscious knowledge about the possible solution to the task. However, the more narrow the realized niche the more relative descriptive value have the available cases used for testing and tuning the system. In the limit, if the realized niche was just a single case, knowing the output value for that case would be obviously sufficient.

4 Recipes

Being eminently inductive, the social robot design process can be seen as analogous to machine learning or inductive modeling in general, only that it is the designer (and not the machine) who looks for an appropriate hypothesis (basically one that works well in the cases that he/she tests). In any case the objective of such inductive process is to have good performance for unseen cases.

In machine learning the training set is used to seek an hypothesis. Thus, the cases used for testing the working hypothesis in the robot development process would be the counterpart of the training set in machine learning.

In machine learning, we have seen that when there is little available knowledge about the solution, the hypotheses to consider may be arbitrarily complex (by contrast, when we have much knowledge the hypotheses that we have to consider are simpler). Complex hypotheses may imply overfitting. In such cases good performance may still be obtained, but only for few specific cases of the domain. This problem would also exist in the robot design process: we would be measuring low error for the tested cases but the error would be large in future, unseen cases.

In machine learning there is a principled way to obtain hypotheses that do not overfit: complexity penalization. A well-known complexity penalization technique is Structural

Risk Minimization (SRM) [4]. SRM is a procedure that considers hypotheses ranging from simple to complex. For each hypothesis the error in the training set is measured. The best hypothesis is that which minimizes the sum of a measure of its complexity and its error in the training set. In other words, it is the simplest hypothesis that gives an acceptable error figure in the training set.

The same idea could be applied in the broader context of the robot engineering design process. Given the fact that for our problem we have little knowledge about the solutions (this was seen in Section 2), a principled way to search for hypotheses would be to start from simple implementations and proceed to more complex ones, each implementation being thoroughly tested in the robot niche. The best implementation should be the simplest one that achieves an acceptable error rate in the robot niche.

Note that this is not the same as saying that simpler techniques should perform better than more complex ones. The point is that complex implementations should be carefully adapted to the realized niche. The best way to achieve this is to start from simple implementations. Using an analogy, it would be like a custom suit. The tailor starts by using a standard pattern of sizes (which fits everybody relatively well) and then, through successive fittings with the client, adjusts different parameters to produce a better final product. In the end the suit will fit the client very well, although it will probably not be adequate for other clients.

There is another recent Machine Learning paradigm that aims at reducing overfitting. It has been recently shown that the concept of *stability* is central to building effective learning systems. A learning algorithm is said to be stable if the deletion (or addition) of one training sample does not change much the learned hypothesis. It has been shown that algorithms that are stable generalize[1] [13]. The concept of stability essentially states that the search process should proceed by incremental changes. Thus, for the case of the robot design and development process, the search for solutions must proceed in an incremental way. New implementations should gradually improve and expand previous ones. That would be an appropriate approach to the problem.

After reviewing the specificities of the problem, at this point we can give a few guidelines that may help to develop sociable robots, in what we may call an opportunistic synthetic approach:

- Make an effort in discovering opportunities for improving performance in the niche of the robot. This can lead to unintuitive implementations (or, in other words, can call for originality).
- Proceed from simple to complex algorithms and representations. The final implementation should be as simple as possible. Implementations should improve iteratively.
- Treat available human knowledge very cautiously, and always following the two previous guidelines. Basic knowledge will almost always be applicable to most niches. Detailed knowledge may be appropriate only for few specific niches.

[1] Note that generalization is the opposite of overfitting. An algorithm generalizes when its performance measured in the training set is indicative of the expected performance for future cases.

This approach basically advocates for a simple-to-complex development process, with emphasis on extensive tests in, and fit to, the robot niche. Note that this can be achieved either from a design or developmental perspective. Using genetics terminology, such adaptation and to the robot niche and incremental competence can be achieved through ontogenetical or phylogenetical approaches. In the former, the robot itself would acquire novel abilities through strong interaction with its environment and/or caregivers, which is essentially what humans do.

5 CASIMIRO

Following the approach mentioned in the previous section, we are building a robot with basic interaction abilities named CASIMIRO, see Figure 1. When finished, the robot is expected to have minimum two-way capacities for engaging in relatively natural social interactions. CASIMIRO is a robotic head with an external appearance similar to Kismet. Currently, the robot has abilities for detecting and tracking people, localizing sound, talking to people, etc. For space reasons we do not describe here all the abilities implemented in the robot, those details can be found elsewhere [8,9]. As an example of the use of the proposed approach we will only explain the owner recognition capacity, which we believe is representative.

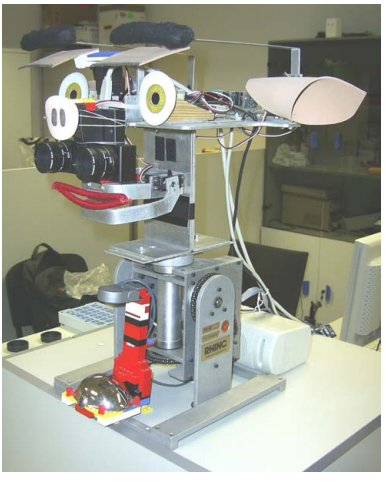


Fig. 1. CASIMIRO

CASIMIRO can recognize its owner or caregiver. In order to recognize a person, the first technique that comes to mind is face recognition. The first author has some experience with the technique, and it was considered that it is not as robust as desired for using it in the robot. Instead, we decided to take advantage of the robot niche.

We are interested in recognizing the person who cares the robot and uses it more frequently (this person is generally the owner too). This person is the one who switches the

robot on, which is usually done from a certain part of the robot or from a certain computer. In the case of CASIMIRO, the main computer (from which the robot is switched on) is situated behind the robot, on the same table.

A camera was placed on top of the robot's main computer monitor. The owner detection module uses that camera to search for a skin coloured blob in the image. The camera has a wide-angle lens, and a relatively low resolution of 160x120 is used. When no blob is encountered the module notifies the robot attention module of that event. At that moment the module exits in order to free CPU resources.

Once it has been notified by the owner detection module, the attention module considers the owner as the first blob that "enters" the interaction space from the left. From that moment on, the "owner" property is stored along with the individual in the person tracking process. With that property the robot can talk to the person as its owner or caregiver.

Face recognition researchers tend to measure performance in terms of the number of individuals that the system can recognize and *measured* error rate. A *measured* error rate of 5-10% can be considered very good. Our implementation is an example of what can be obtained by following the general approach mentioned above: although it is only appropriate for the current robot niche, we have been able to devise the simplest algorithm (or one of the simplest) that allows to recognize the owner. Thus, despite its simplicity, or rather because of it, our implementation can recognize the owner/caregiver with *guaranteed* zero error. In preliminary experiments, people received such ability with amazement, which in turn was exciting to us.

In experiments, people received such ability with amazement, which in turn was exciting to us. Some asked questions like *"how can the robot do that?"* or *"hey! does it recognize people?"*. We arranged for some people to interact with the robot and fill in a questionnaire after the interaction. In each session the owner entered the interaction zone after a while (so that from that moment on there were two people in front of the robot). One of the questions was "The robot recognizes people". People had to assign it a score of 1 (definitely no) to 5 (definitely yes). After 19 interviews (37.5% of the interviewees had a PhD in computer science or an engineering), the result was that the mean score assigned to this question was 3.58 (standard error of 0.31). We find these results encouraging, given the simplicity of the implemented technique.

6 Conclusions

Sociable robots are being actively researched in the robotics community. Such robots purportedly implement some of the human social abilities. The interest is more than justified, for social intelligence is in fact one of the main constituents of generic human intelligence.

In this position paper it has been shown that the reproduction of social intelligence, as opposed to other types of abilities, may lead to overfitting: good performance in tested cases but poor performance for future (unseen) cases and situations. It has been argued that this limitation stems from the fact that social abilities are mainly unconscious to us. Given such limitation, a set of basic guidelines has been proposed for developing sociable robots. These are being put into practice in a robotic head named CASIMIRO.

The utility of the interpretation and specific guidelines proposed here for developing sociable robots shall have to be explored in future projects. Of course, it is not intended to be a definitive approach for developing sociable robots. It proposes just a few non-stringent rules, and hence it may be useful as a basic development perspective, possibly in conjunction with other paradigms and approaches. We hope that it will be present in our future robots and in those built by other researchers. Only a large scale use of the approach will allow to extract meaningful conclusions.

Acknowledgments

This work was partially funded by research projects *UNI2005/18* of Universidad de Las Palmas de Gran Canaria and *TIN2004-07087* of the Spanish Ministry of Education and Science and FEDER.

References

1. Baars, B.J.: A cognitive theory of consciousness. Cambridge University Press, New York (1988)
2. Breazeal, C.L.: Designing social robots. MIT Press, Cambridge, MA (2002)
3. Bullock, D.: Seeking relations between cognitive and social-interactive transitions. In: Fischer, K.W. (ed.) Levels and transitions in children's development: new directions for child development, Jossey-Bass Inc. (1983)
4. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
5. Cleeremans, A.: Conscious and unconscious processes in cognition. In: Smelser, N.J., Baltes, P.B. (eds.) International Encyclopedia of Social and Behavioral Sciences, vol. 4, pp. 2584–2589. Elsevier, London (2001)
6. Dautenhahn, K.: Getting to know each other - artificial social intelligence for social robots. Robotics and Autonomous Systems 6, 333–356 (1995)
7. De Renzi, E.: Prosopagnosia. In: Feinberg, T.D., Farah, M.J. (eds.) Behavioral Neurology and Neuropsychology, McGraw-Hill, New York (1997)
8. Deniz, O.: An Engineering Approach to Sociable Robots. PhD thesis, Department of Computer Science, Universidad de Las Palmas de Gran Canaria (July 2006)
9. Deniz, O., Hernandez, M., Lorenzo, J., Castrillon, M.: An engineering approach to sociable robots. Journal of Experimental and Theoretical Artificial Intelligence (in press, 2007)
10. von Helmholtz, H.: Physiological Optics. Optical Society of America (1924)
11. Hutchinson, G.E.: Concluding remarks. Cold Spring Harbor Symp. Quant. Biol. 22, 415–427 (1958)
12. Lindblom, J., Ziemke, T.: Social situatedness of natural and artificial intelligence: Vygotsky and beyond. Adaptive Behavior 11(2), 79–96 (2003)
13. Poggio, T., Rifkin, R., Mukherjee, S., Niyogi, P.: General conditions for predictivity in learning theory. Nature 428, 419–422 (2004)
14. Sokolov, E.N.: Perception and the conditioned reflex. MacMillan, New York (1963)

Processing of Myoelectric Signals by Feature Selection and Dimensionality Reduction for the Control of Powered Upper-Limb Prostheses

Klaus Buchenrieder

Universität der Bundeswehr München, Fakultät für Informatik,
Institut für Technische Informatik IV, D-85577 Neubiberg, Germany
klaus.buchenrieder@unibw.de

Abstract. The extraction of features from myoelectric signals (MES) for the classification of prehensile motions is difficult to achieve. The optimal selection of features, extracted from a MES and the reduction of dimensions is even more challenging. In the context of prosthetic control, dimensionality reduction means to retain MES information, that is important for class discrimination and to discard irrelevant data. Dimensionality reduction strategies are categorized into feature selection and feature projection methods according to their objective functions. In this contribution, we bring forward a statistical cluster analysis technique, which we call the "Guilin Hills Selection Method". It combines selection plus projection and can be applied in the time- and in the frequency-domain. The goal is to control an electrically-powered upper-limb prostheses, the UniBw-Hand, with a minimum number of sensors and a low-power processor. We illustrate the technique with time-domain features derived from the MES of two sensors to clearly differentiate four hand-positions.

1 Introduction

Research in MES processing dates back to the early 1960s [1] and many of the techniques developed in the early years are now transformed into medical products. These benefit largely from progress in microelectronics and computer technology. An essential compendium of advances and a review of the state-of-the-art, was brought forward by N‰der in 1988 [2]. A more recent survey of the progression in upper-limb prosthesis and microprocessor-control, was collocated by Lake and Miguelez [3]. In addition, Muzumdar provides an extensive overview over excitation methods, prosthetic implementations and clinical applications [4]. Today, most control systems are based on Hudgins' theoretical foundation and exert, for example, a simple multilayer perceptron (MLP) as classifier for a set of time-domain features extracted from an MES [5,6,7]. Based on these fundamental ideas, a prosthesis control system is under development at the Universit‰t der Bundeswehr M¸nchen [8]. The UniBw-Hand controller is different from earlier approaches with respect to feature classification, dimensionality reduction, training method and comfort for the user.

For the operation of the system, we distinguish a feature selection-, a training- and an operative-mode. A reliable and repeatable classification of the MES is assured

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1057 – 1065, 2007.
© Springer-Verlag Berlin Heidelberg 2007

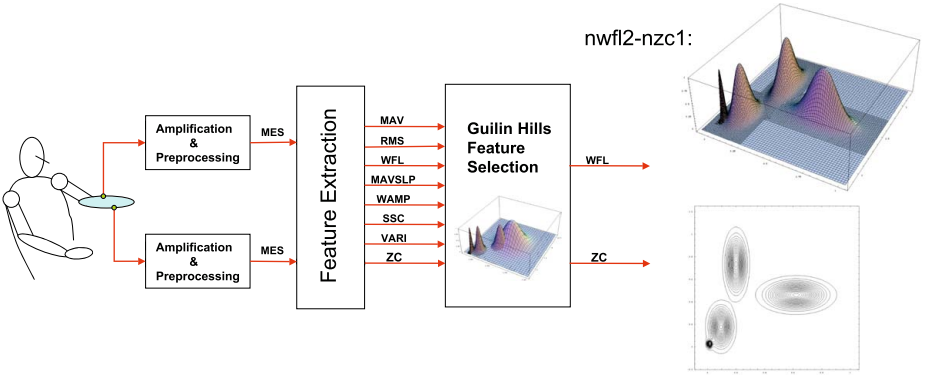


Fig. 1. A subset of features, ensuring highly reliable classification results, as depicted in the surface- and contour-plots to the right of the system chart, are obtained with the Guilin Hills selection method

through the selection of a curtly set of features. The set is calculated from the MES engendered by remaining predominant muscles, e.g. on the patients forearm, for predefined hand and arm motions or positions. With the Guilin Hills selection method, a proper subset is chosen, so that a minimal number of features relates a hand-position to a feature pairing with a high probability. Figure 1 shows the procedure and the result for the feature combination of normalized waveform length for sensor 2 (nwfl2) and normalized zero crossings for sensor 1 (nzc1).

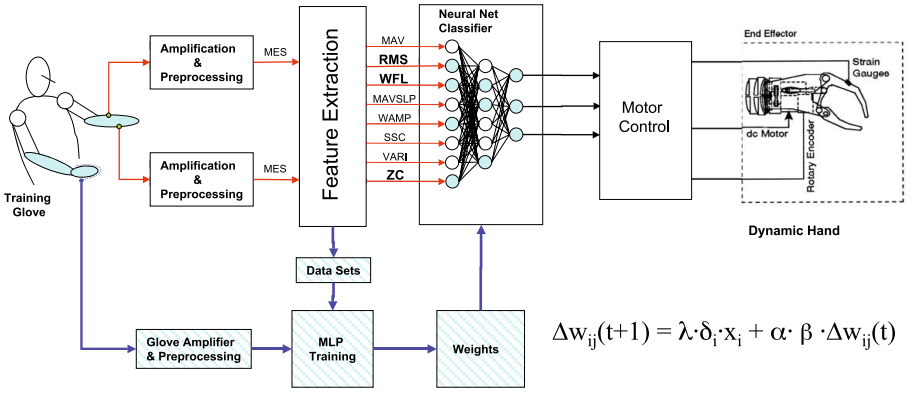


Fig. 2. In the illustration of the UniBw-Hand control system for powered upper-limb prostheses control, shaded training elements and negligible features are removed when the hand is operated

The training of the multilayer perceptron, that classifies the MES in the operative mode, can start as soon as the combination of features is determined. In contrast to traditional training methods, where users are asked to solely flex muscles without feedback, we also measure the precise motion of the opposite hand with a data-glove.

As presented in [8], we can accurately relate hand gestures and movements to the MES of the impaired side with the information gained from the glove. The set of features is then extracted and presented to the neural net classifier. The correlation of movement and MES ensures precise training data and accurate weights for the operative mode.

When the prostheses is used in the operative-mode, the MES is measured and preprocessed exactly as in the training mode. With the weights obtained during the training phase, controls for the motor unit are generated in real-time. Amplifiers of the motor control-unit provide steering signals for the prosthetic hand device or the end-effector. A current limiter for the motor and dedicated sensors provide sufficient feedback to the motor control unit and ensure a force regulated and slippage reduced grip.

2 Dimensionality Reduction

In myoelectric signal processing, dimensionality reduction is the process of feature selection and feature reduction. The goal dimensionality reduction is to retain important information for best class discrimination and to discard superfluous information. Fodor provides a general survey of dimensionality reduction approaches and classifies the methods into feature selection and feature projection techniques [9].

Feature selection methods try to return the best subset of an original, initial set of features. In the context of myoelectric signal processing, feature selection is based on Euclidian distance calculations and a class separation criterion. Supervised methods rank the features according to available class membership information. Generally, the features are chosen, so that the distance of the “centers of mass”, determined for the sample-clouds of hand-positions, obtained during a training phase, are maximized.

Feature selection methods based on the center of mass paradigm become unsuitable, when extensive intersections of sample-clouds exist. Since individual muscles contribute simultaneously to several hand-motions or -positions, a sample point, as found in the strongly clustered center of a cloud of mass points, classifies a hand-position better than a more distant point. As an example, regard the flexor carpi radialis muscle. It is flexed when adopting a fist, a radial flexion and other positions. When all points contribute to the center calculation with identical weights, outlier points shift the center of gravity away from the region for an optimal classification. Therefore, features must be chosen, so that the centers of probability, the hills, become situated with maximal distance.

Feature projection methods attempt to determine the best combination of original features through a mapping of the original multidimensional feature space into a space with fewer dimensions. Typically, an original feature space is transformed with a linear transformation, e.g., with principal component analysis (PCA), that provides a means of unsupervised dimensionality reduction, as no class membership qualifies the data when specifying the eigenvectors of maximum variance [7].

3 The Guilin Hills Selection Method

This method was inspired by the beautiful landscape of the hills and the Li river close to Guilin in southern China, and the observation, that sets of feature-data points are normally distributed. This has been confirmed by testing several hundred sets of training data from three individuals, which were all found to be mesokurtic, with $\Upsilon_N \approx 0$. This observation allows us to represent a set of training data for a feature of a specific muscle and a certain hand-position as a tuple consisting of the mean (a_n) and the standard deviation (d_n).

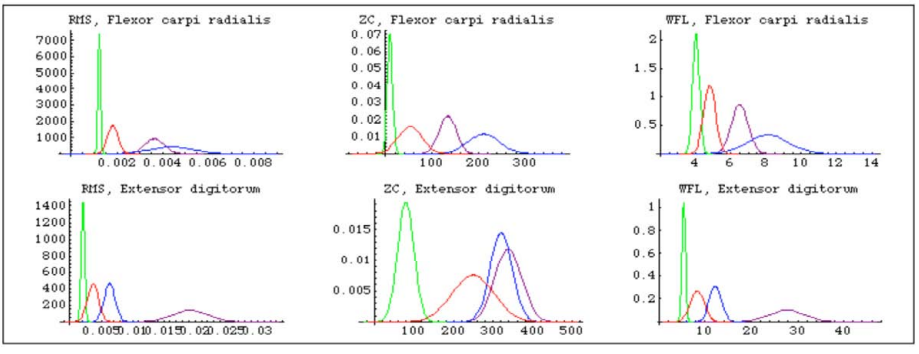


Fig. 3. Display of four hand-positions over three features and two muscles

The bell-shaped curves in each individual illustration of Figure 3 show four distinct hand-positions: radial flexion, wrist-extension, fist, and wrist-flexion. Initially, we calculated all nine standard features and selected, for this disquisition with the method introduced, the features: root mean square (RMS), zero-crossings (ZC) and wave-form length (WFL). These features were derived for the flexor carpi radialis and the extensor digitorum muscles from sets of 100 training samples each. The RMS is a statistical measure for the magnitude or strength of the muscle contraction and the wave-form length provides information about the frequency range and the amplitude of the signal. The number of sign-changes within a predefined time-window, are denoted as zero-crossings and provide information about frequency related properties.

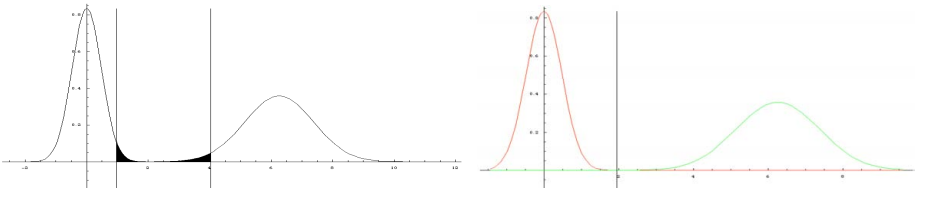


Fig. 4. The left graph (a) depicts the combined 3σ areas, the grey vertical in (b) marks the intersection point x_{int}.

During the operative phase of the prostheses, continuously extracted features, called measurements, are derived from the MES and classified based on the training data. An ideal association with a position is possible only when the probability density functions are well separated. For this reason, features must be chosen, so that either: the integral under the combined curve is minimal within, i.e. three, standard deviations of the respective means, Figure 4a; or the combined density function at the intersection is minimal, as in 4b. The combined density function at the intersection point x_{int}, for two feature tuples (a_1,d_1) and (a_2,d_2), simply calculate as:

$$CombinedNvd\left(x_{\text{int}+},x_{\text{int}-},a_1,d_1,a_2,d_2\right)=\max\left(\begin{array}{cc}\dfrac{1}{\left(\sqrt{2\pi}d_1\right)e^{\frac{\left(x_{\text{int}+}-a_1\right)^2}{2d_1^2}}}+\dfrac{1}{\left(\sqrt{2\pi}d_2\right)e^{\frac{\left(x_{\text{int}+}-a_2\right)^2}{2d_2^2}}}, \\ \dfrac{1}{\left(\sqrt{2\pi}d_1\right)e^{\frac{\left(x_{\text{int}-}-a_1\right)^2}{2d_1^2}}}+\dfrac{1}{\left(\sqrt{2\pi}d_2\right)e^{\frac{\left(x_{\text{int}-}-a_2\right)^2}{2d_2^2}}}\end{array}\right); \tag{1}$$

with

$$Intersection\left(a_1,d_1,a_2,d_2\right)=x_{\text{int}+},x_{\text{int}-}=\frac{1}{d_1^2-d_2^2}\left(\begin{array}{l}a_2d_1^2-a_1d_2^2\pm \\ \sqrt{\left(\begin{array}{l}a_1^2d_1^2d_2^2-2a_1a_2d_1^2d_2^2+a_2^2d_1^2d_2^2- \\ 2d_1^4d_2^2\log\left(\dfrac{d_2}{d_1}\right)+2d_1^2d_2^4\log\left(\dfrac{d_2}{d_1}\right)\end{array}\right)}\end{array}\right).$$

Obviously, only one of the two resulting intersection points is valid, because it is situated between the "hills" and can be used for feature selection. In the remaining presentation, we refer to the combined density at the intersection point simply as crossover. Even though features overlap, the zero crossing (ZC) feature for the flexor carpi radialis provides the best differentiation, as the example in Figure 3 shows. The result can be improved only by adding a carefully selected, additional feature as second dimension, resulting in a two-dimensional Gaussian function:

$$f(x,y)=Ae^{-\left(t_{11}\left(x-x_0\right)^2+t_{12}\left(x-x_0\right)\left(y-y_0\right)+t_{22}\left(y-y_0\right)^2\right)}; \tag{2}$$

Where the matrix

$$\mathrm{t}=\begin{bmatrix}t_{11} & t_{12}\\ t_{21} & t_{22}\end{bmatrix}=\begin{bmatrix}\left(\dfrac{\cos\varphi}{d_x}\right)^2+\left(\dfrac{\sin\varphi}{d_y}\right)^2 & \dfrac{\sin 2\varphi}{d_y^2}-\dfrac{\sin 2\varphi}{d_x^2}\\ \dfrac{\sin 2\varphi}{d_y^2}-\dfrac{\sin 2\varphi}{d_x^2} & \left(\dfrac{\sin\varphi}{\mathrm{d_x}}\right)^2+\left(\dfrac{\cos\varphi}{d_y}\right)^2\end{bmatrix};$$

Is positive-definite

In Equation (2), A represents the amplitude and (x_0, y_0) determines the center of the hill. The angle φ, by which the hill is rotated, results from the linear fitting of the training-data for a certain combination of features. As an example, consider Figure 5a. It depicts the hill for a fist-position calculated for the features RMS of the flexor carpi

radialis and the WFL for the extensor digitorum muscle. Figure 5b visualizes the source data, a set of training samples, for which the angle φ was calculated by linear line fitting. Note, that the sample points were selected with an unusually wide spread.

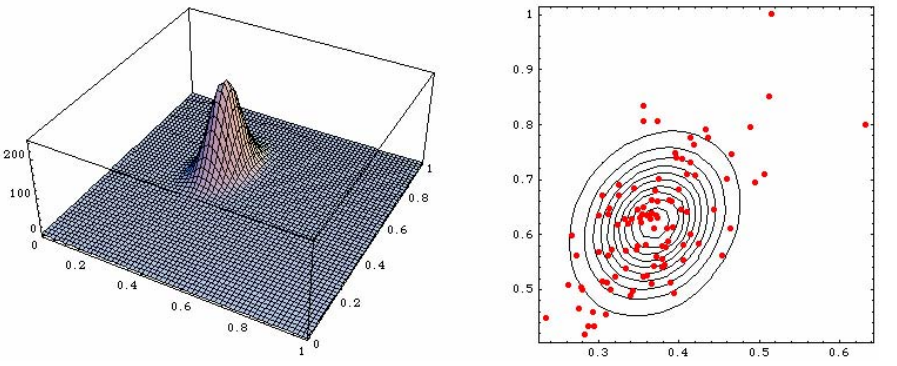


Fig. 5. The left subfigure (a) shows the two-dimensional probability function for a single hand-position over the features RMS and WFL. The contour plot (b) indicates the alignment of the hill according to the training data, yielding φ.

When more than two positions are to be distinguished, the mutual distance between all hills must be as large as possible. Only a maximum distance ensures, that measured points can be properly classified and the hand-positions clearly identified. Since the combination of features determines the separation of the hills, one must calculate the largest, so called worst-case crossover for all combinations of features with sensor pairings. The desired combination is the pairing with the lowest value for the worst-case crossover. Figure 6b shows the well separated elliptic footprints of the perpendicular aligned hills in 6a. The cross-sectional cut, as required to determine the crossover, is shown in 6c.

$$x = d_1 \cos\varphi; \quad y = d_2 \sin\varphi; \Rightarrow d_{csl} = \sqrt{\frac{d_1^2 d_2^2 \sec\varphi^2}{d_1^2 + d_2^2 \tan\varphi^2}}; \tag{3}$$

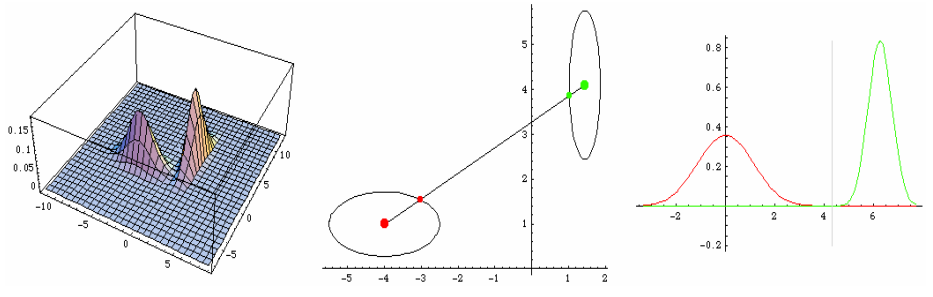


Fig. 6. Two hand-positions over two standard features (a), 3σ contour-plot with a cross sectional cut-line (b), constructed as baseline for the combined density function in (c).

The standard deviations d_{cs1}, d_{cs2} are derived with the parametric Equation (3). The Euclidian distance of the points, where the semi-minor and semi-major axes of the ellipsis intersect, serve as the new means a_{cs1} and a_{cs2}.

4 Experimental Results and Concluding Remarks

The Guilin Hills feature selection method was tested with several hundred sets of training- and measurement-data from three individuals. For data recording, we used four active, bipolar skin-surface electrodes and a separate common ground electrode (total gain G_{12}=200, pass-band window 70 – 400 Hz). The four channels were sampled with a PCIe-6251 Express M Series multifunction DDA board from National Instruments with an accuracy of 16 Bits and a sampling rate of 1024 Hz. From the three second sampling intervals, we extracted the standard time-domain elements: root mean square (RMS), zero crossings (ZC), mean absolute value (MAV), waveform length (WFL), mean absolute value slope (MAVSLP), Willison amplitude (WAMP), slope-sign change rate (SSC), integrated absolute value (IAV) and the variance of the EMG (VAR). With the Guilin Hills selection method, implemented with Mathematica 5.2, we reduced the set of time-domain features to RMS, ZC and WFL for two channels, corresponding to the flexor carpi radialis and the extensor digitorum muscle. Previous work [10,11,12] shows, that the selected features are suitable for our experimental UniBw Hand controller. For the hand-positions: radial flexion, wrist-extension, wrist-flexion and fist, the Guilin Hills selection method returns the feature selection matrix given in the Table below.

Table 1. Feature selection matrix: RMS-WFL-ZC-42

RMS2	WFL1	WFL2	ZC1	ZC2	
1.3952	.258550	.0201346	.00440842	.0210817	RMS1
	.208677	.0501229	.00391864	.00391293	RMS2
		.0405795	.00352604	.02322640	WFL1
			.00335150	.00446167	WFL2
				.00453696	ZC1

The minimal crossover of 0.00335150 corresponds to the feature combination ZC1-WFL2, which provides the best separation for two sensors. The Guilin Hills surface plot for this combination is depicted in Figure 7a. The second best solution requires only one physical sensor and yields a worst-case crossover of 0.00352604.

The surface plot for the combination ZC1-WFL1, exhibiting a remarkable separation of the positions, is shown in Figure 7b. Whether a commercial prosthetic device, such as the MYOBOCK™ System Electrohand from Otto Bock Healthcare GmbH, can be controlled in a satisfactory, reliable and safe manner with a minimum number of sensors is part of the ongoing research.

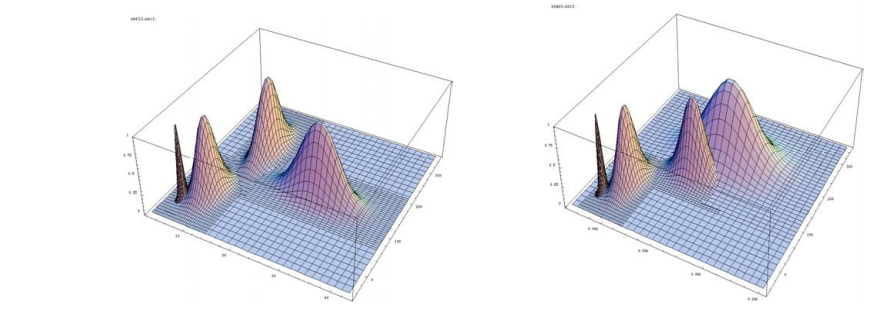


Fig. 7. Normalized results for the feature combinations: (a) ZC1-WFL2 and (b) ZC1-WFL1

Even though, the presented Guilin Hills feature selection method was found to significantly improve the classification of MES over established methods, we found, that future research requires a higher quality of the recovered MES. For this reason, current work includes the development of autocorrelation filters to be integrated in the first-stage of the sensor electronics and the development of multi-electrode arrays. Improvements of the classification stage have been achieved with a new “spyglass” technique. This technique allows us to switch between feature combinations when a measurement cannot be properly classified due to a large crossover.

References

1. Horn, G.: Electromyographic signal produced by muscle movement, controls grasp of prosthetic fingers. In: Electronics, pp. 34–36 (October 1963)
2. Nder, M. (ed.): Prothesen der oberen Extremitt. 1. Seminar am 19. Mrz 1988 in Mnchen, Otto Bock Stiftung. Mecke Druck und Verlag, Duderstadt (1988)
3. Lake, C., Miguelez, J.: Comparative Analysis of Microprocessors in Upper Limb Prosthetics. American Academy of Orthotists and Prosthetics Journal of Prosthetics & Orthotics 15(2), 48–65 (2003)
4. Muzumdar, A.: Powered Upper Limb Prostheses. Springer, Berlin (2004)
5. Hudgins, B.: A novel approach to multifunctional myoelectric control of prostheses. Ph.D. Thesis, Department of Electrical Engineering, University of New Brunswick, Frederiction, New Brunswick, Canada (1991)
6. Hudgins, B., Parker, P., Scott, R.: A New Strategy for Multifunction Myoelectric Control. In: IEEE Transactions on Biomedical Engineering, vol. 40, pp. 82–94. IEEE Computer Society Press, Los Alamitos (1993)
7. Englehart, K., Hudgins, B., Parker, P., Stevenson, M.: Classification of the Myoelectric Signal using Time-Frequency Based Representations. Special Issue of: Medical Engineering and Physics on Intelligent Data Analysis in Electromyography and Electroneurography (1999)
8. Buchenrieder, K.: Processing of Myoelectric Signals for Control Assistance of Multifunctional Hand Prosthesis. In: Proceedings of the International Mediterranean Modelling Multiconference, Piera, LogiSim, Barcelona, Spain, pp. 653–657 (September 2006)

9. Fodor, I.: A Survey of Dimension Reduction Techniques. Report of the U.S. Department of Energy. Report Nr. UCRL-ID-148494, Lawrence Livermore National Laboratory, May (2002)
10. Sippel, M.: Verarbeitung myoelektrischer Signale zur Erkennung von Bewegungen und Stellungen einer Hand. Diplomarbeit, Fakultt fr Informatik, Universitt der Bundeswehr Mnchen, UniBwM-ID 40/2005, Neubiberg, Germany (2005)
11. Trautsch, R.: Biosignalverarbeitung zur Steuerung einer Handprothese. Diplomarbeit, Fakultt fr Informatik, Universitt der Bundeswehr Mnchen, UniBwM-ID 39/2005, Neubiberg, Germany (2005)
12. Buchenrieder, K.: Dimensionality Reduction for the Control of Powered Upper Limb Prostheses. In: Proceedings of the 14th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, Tucson, Arizona, pp. 327–333 (March 2007)

The Anglet Experiment: A Cybercar on the Beach

Joseph Canou, Damien Sallé, Marc Traonmillin, and Vincent Dupourqué

ROBOSOFT
Technopole d'Izarbel, 64 210 Bidart, France
{joseph.canou, damien.salle, marc.traonmillin,
vincent.dupourque}@robosoft.fr

Abstract. This paper presents the "robuCAB", a Cybercar developed by ROBOSOFT, and the results of experimentation in a public and pedestrian area. In a first part, the hardware and software architecture of the vehicle - control, safety and HMI - are presented. The results of experimentation are detailed and the perspectives of evolution (control, safety, HMI) are discussed.

Keywords: Cybercar, control architecture, safety, HMI, experimentation.

Introduction

This article presents the results of an experimentation carried out within a French project called MobiVIP[1] (funded by the French Research Ministry). This experimentation concerns the circulation of a cybercar (robuCAB, developed by Robosoft) in a pedestrian environment. The aim of this experimentation was the validation of the technical and safety aspects of the vehicle. It concerns also the acceptance of the use of a cybercar by the public. This 4 weeks experimentation allowed the transportation of approximately 500 persons.

Robosoft supplies advanced mobile robotics solutions to drastically reduce the cost of services in transport, thanks to its line of mobile robots, its embedded technology of control systems, and its expertise demonstrated in various fields of service robotics since 1985, including research. Robosoft's core business is the automatic transport of people and goods in traffic-less areas. They have already designed and developed several cybercars:

- Cycab in collaboration with INRIA. Cycab is a cybercar allowing the transportation of 2 persons,
- RobuRide which is an autonomous system dedicated to people transportation. It is used in a historical touristic area (Maginot Line in France),
- RobuCAB, a cybercar allowing the transportation of 4 persons in an automated way.

[1] http://www-sop.inria.fr/mobivip

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1066–1072, 2007.
© Springer-Verlag Berlin Heidelberg 2007



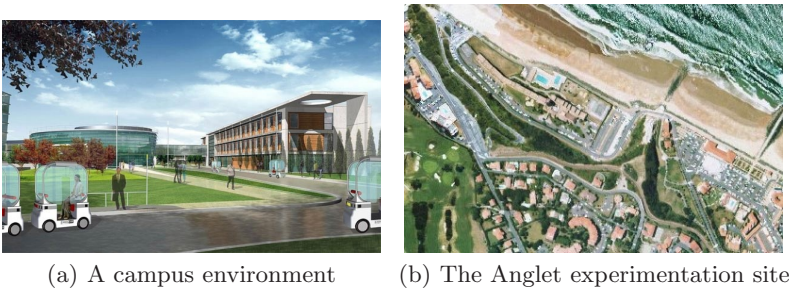
(a) In a technopole environment (b) Providing service along the beach

Fig. 1. The robuCAB vehicle

A cybercar is a vehicle dedicated to people and goods transportation in a fully automated way, as illustrated in figure 1.

These future transport modes apply to all the protected sites receiving a high concentration of people having to move on relatively short distances (few kms) in indoor or outdoor environments, as illustrated in figure 2. These vehicles work then either in standalone vehicle or in a fleet of vehicles managed by a supervisor. Among the protected places being able to receive cybercars one can quote:

- Inner city centers
- Industrials or academicals campus
- Public parks
- Airports



(a) A campus environment (b) The Anglet experimentation site

Fig. 2. Illustration of Cybercars in a dedicated area and the experimentation site

In the future, one can imagine than there will be cybercars able to move at high speed on dedicated zones others than the reduced area than we quoted.

1 RobuBOX Overview

All Robosoft's robots are provided with the robuBOX[2]: a generic and advanced robotics controller software, developed with Microsoft Robotics Studio and

[2] `http://www.robosoft.fr`

implementing both low level control and high-level mobile robotic functions like path recording and following, obstacle detection, localization and basic navigation... It allows to easily and rapidly generate Service Based Architectures dedicated to service robotics.

The robuBOX main target is to provide researchers, integrators and manufacturers with an off-the-shelf solution to quickly and easily build standalone or fleets of service robots, such as AGV (Automatic Guided Vehicles), scrubbing machines, manipulators, cybercars, and so on. Based on reference designs for both the hardware platforms and control software, and thanks to the robuBOX, it becomes really easy and fast to transform any machine into a professional service robot.

The Robosoft's robuBOX software is really open for researchers through at least three important features:

- it provides existing services and control architectures that can be completely modified: one can develop his own service and integrate it in one of proposed architectures, or he can experiment completely new architectures using existing services as well as third parties services
- To allow re-use of services and easy integration of new services in existing architectures, Robosoft has proposed definitions for standardized interfaces between robotic components and algorithms. Using these generic contracts, one can really easily develop a new algorithm and still use all the remaining robuBOX functionalities and provided architectures to deploy, test and evaluate it.
- It is not limited to the use of a specific OS, as gateways can be easily implemented to communicate information between services and other microcontrollers or CPUs (potentially running other OS). This can be achieved, among others, through TCP, serial links, webServices or other solutions.

The robuBOX, as it is Service based is really well suited to collaborative developments : once the goals and interfaces of each service have been defined, each user can carry out developments on his own. Once all the services are put back into the robot control system, the hassle of software integration is greatly reduced.

Another very important feature of robuBOX that is the realistic simulation environment: every robot from Robosoft is provided with its realistic model in the simulation (both physical/dynamic properties and graphic properties), as illustrated in the following pictures. The major benefit of the robuBOX simulation and generic interfaces is that the exact same service architecture can be used indifferently on the real robot or in the simulation. This has very strong consequences:

- The users can work on the development of their own services using the simulations: they do not need a real robot in the first phases of a project.
- There is a drastic gain in time and development effort for the prototyping and implementation of robotic control architectures, as they can be optimized while the robots are being built.

– Experimental testing is faster while still as efficient as when using real robots since the control architecture remains exactly the same between the simulation and the real robot.

The robuBOX, as it is Service based is really well suited to collaborative developments: once the goals and interfaces of each service have been defined in early phases of a project, each partner can carry out developments on his own. Once all the services are put back into the robot control system, the hassle of software integration is greatly reduced.

2 Description of the Anglet Experiment

The experimentation carried out within a French funded project called MobiVIP concerns the circulation of a cybercar (RobuCAB, developped by Robosoft) in a pedestrian environment. The aim of this experimentation was the validation of the technical and safety aspects of the vehicle. It concerns also the acceptance of the use of a cybercar by public in an open environment for a long period.

The objectives of this experimentation were as follows:

– Validation of the vehicle behavior in a crowd
– Validation of user acceptance
– Get feedback about user interface and possible vehicle improvements
– Feedback about safety issues

Description of the frame of experimentation:

– Duration : 4 weeks
– Path is a 700m loop with 4 possible stops
– The width of the path followed by the vehicle is 3m wide
– Localization made with a DGPS and odometry

The whole trajectory followed by the vehicle is composed by several sections. The sub-trajectories are learned using a joystick (teach by showing process). During this teach by showing process the characteristic points of the path are obtained from the DGPS system. The user defines then a scenario from the network of pre-learned sub-trajectories.

The figure 3 illustrates a teach by showing process.

The figure 4 is a screenshot of the HMI available to the user of the robuCAB. On a touch screen he can choose the starting and ending point of his course and visualize the result on a real picture of the site.

3 Autonomous Navigation of the RobuCAB

This navigation system is divided in 2 parts (see figure 5) :

– Localization: the information is coming from DGPS and Wheels sensors. They are combined to obtain reliable position/orientation information.

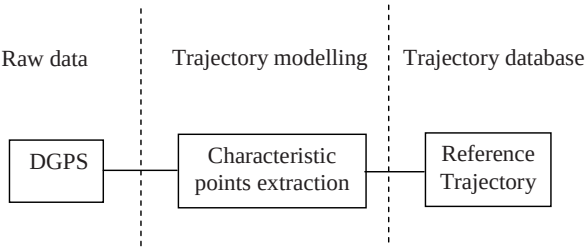


Fig. 3. Teach by showing process

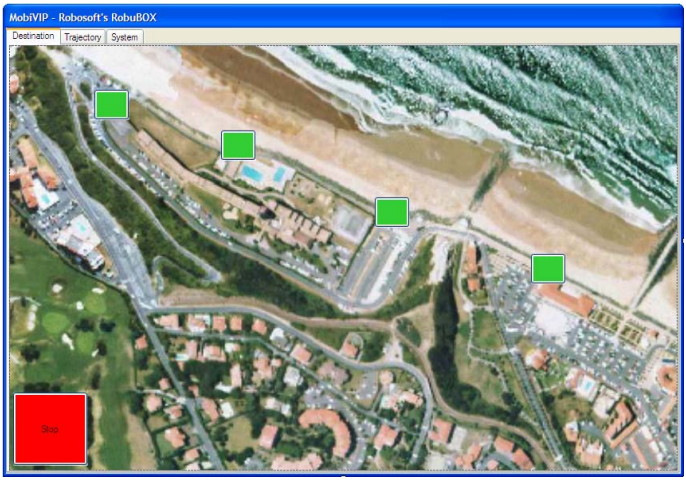


Fig. 4. HMI embedded in the robuCAB during the Anglet experiment

– Control law: current position from localization is compared to the reference trajectory. This comparison allow us to generate the speed and steering control of the vehicle allowing the trajectory following.

The aim of the localization system is to perform the low acquisition rate of the DGPS and some lack in the data. Each time a valid DGPS data is received the wheel sensors position estimation is reevaluated based on the DGPS information.

The robuCAB navigation is made using a control which combines orientation error and lateral error with respect to a reference trajectory (see figure 6).

The control law implemented on the vehicle is as follows:

$$SteeringAngle = D \times K_{pp} \times ErrorSteer \times K_e \times \frac{180}{\Pi} + OffsetSteer$$

In this control law the parameters are:

– D: lateral error (orthogonal projection on the followed line)
– $ErrorSteer$: steering error

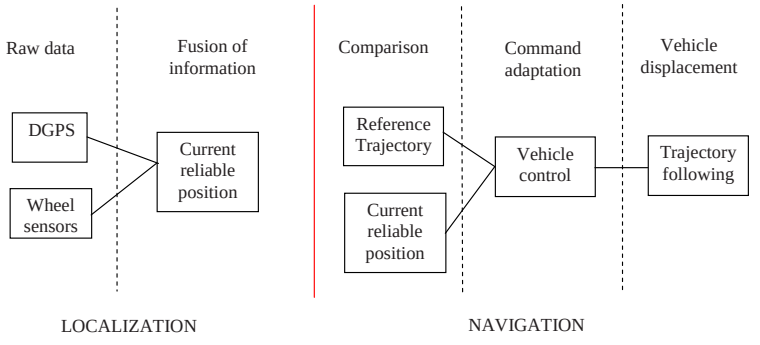


Fig. 5. The autonomous navigation process

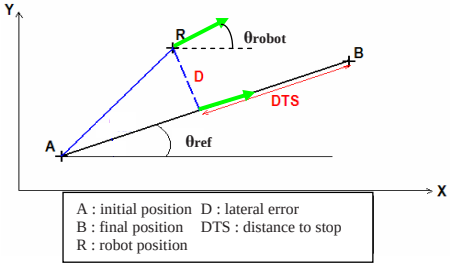


Fig. 6. Control law

- *OffsetSteer*: mechanical offset
- K_{pp}: proportional gain on lateral error
- K_e: proportional gain on steering error

During the experimentation out of scope of Anglet demonstration, the vehicle is able to follow a trajectory up to 18km/h. For safety reasons (a lot of pedestrian on the demonstration site, elderly people...) the vehicle speed was limited to 11km/h. This speed was customizable by the user between 1, 2 and 3m/s.

4 Safety

During the displacement of the vehicle, the pedestrian protection is made autonomously using US sensors and a laser range finder.

The pedestrian protection is made through a division of the space in front of the vehicle. Each zone is attached to risk value, this value being dependant on the presence of an obstacle. The speed of the vehicle is managed directly by these risk zones and their values. When an obstacle is detected in front of the

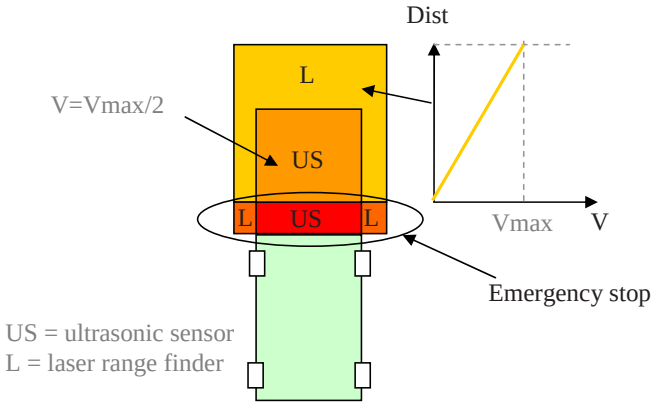


Fig. 7. Safety zone

vehicle (see figure 7), a sound is emitted by the robuCAB to inform the person who could be there.

The vehicle is also equipped with emergency stop button located inside and outside the vehicle. In order to allow the passenger of the vehicle to adapt the vehicle speed to its desires, the HMI offers the possibility to limit the maximum speed of the vehicle.

5 Conclusion

During the 4 weeks of the experimentation, approximately 500 persons used the robuCAB without any major incident. The only encountered problem was material problem: DGPS failure, brake/motor failure and tyres. The comportment of the vehicle was conform to the implemented control law. However, the sensors used for the safety of the vehicle while navigating in crowd do not allow "intelligent" obstacle avoidance. In some case we have been confronted to a static obstacle (like a bicycle parked on the vehicle trajectory) and the vehicle just stops. If we want to improve the behavior of the vehicle in such a case we need more information from sensors. One of the most important information could come from a video system, allowing a more "intelligent" interpretation of the situation. We had also the opportunity to discuss with a person from a French institute working on certification of such a system. From this discussion we obtained information concerning the improvement we could make to our vehicle for utilization in a taxi-like mode, like an automated door closing or a vocal warning instead of a "simple" sound warning. From a general point of view the user acceptance was good. People would like to have such a type of taxi-like transportation system. They were very interested in the ecological aspect of the vehicle and also in the possibility to share a vehicle for short courses, like in inner city.

Crossroad Cooperative Driving Based on GPS and Wireless Communications

José E. Naranjo, Ricardo García-Rosa, Carlos González, Teresa de Pedro, Javier Alonso, and Javier Vinuesa

Instituto de Automática Industrial (CSIC), Ctra. Campo Real, km. 0,200
La Poveda – Arganda del Rey, 28500, Madrid, Spain
{jnaranjo, ricardo, gonzalez, tere, jalonso, jvinuesa}@iai.csic.es
http://www.iai.csic.es/autopia

Abstract. Autonomous vehicles have the capacity of circulating much in the way humans drive them, but without the inherent limitations of people driving. A second step in the development of these kind of vehicles is to add the capacity to perform cooperative driving with other cars to them, autonomously as well manually driven. The aim of this paper is to describe a new kind maneuvers for autonomous vehicles: intersection dealing capability. In our work with autonomous vehicles, we have developed an automatic driving system based on artificial intelligence, with the ability to perform automatic driving in stand-alone routes as well a limited cooperative driving, with some maneuvers controlled: Adaptive Cruise Control (ACC) and overtaking. In this paper, we develop an extension of the cooperative driving system, adding the capacity to deal with road intersections to the cybernetic driver. We have defined a set of cases of use and we have implemented a first prototype of the simplest one, that is the crossing of a priority road. Some experiments have been carried out, using our mass produced autonomous vehicles, equipped with onboard computers, GPS receivers and wireless networking.

Keywords: Autonomous vehicles, fuzzy logic, intersection management.

1 Introduction

In the year 2005, 41,600 people died in road traffic accidents in the European Union. Some 1.9 million people were injured, some of them severely. The economic damages generated by traffic accidents were estimated at €200 billion, corresponding to approximately 2% of the European Union's Gross National Product. In order to solve this problem, European Commission has taken the challenge of reducing by one half this cipher by the year 2010, mainly applying new information and communication technologies. One of the most dangerous maneuvers is the circulation through road intersections and the various modalities of priority and directions. In the USA, 8,703 accidents were happened in intersections in the year 2005, causing 26,734 victims (U.S. DoT Fatality Analysis Reporting System (FARS)).

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1073 – 1080, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The research on intelligent vehicles for intersection management is actually a technological challenge, with some groups working in this area worldwide. The philosophy is the integration of vehicle-infrastructure components and functions into cooperative intersection collision avoidance systems (CICAS, US DOT CICAS Program, http://www.its.dot.gov/cicas), using wireless communication technology. Some developments have been carried out as driving aids for augmenting the safety in roadway intersections. In California PATH Program some Intersection-Decision-Support systems have been developed in order to advise the driver in one of the most critical situations: left turn across path with incoming vehicles [1], and some working scenarios to test these systems have been defined [2]. More USA research are described in [3]. In Europe, several projects of the 6th FrameWork Program (FWP) deal with these driving aids. That is the case of Intersafe Project, where an ADAS is under development to detect a potentially dangerous situation in road intersections and to warn the driver [4].

A second step in the road intersection technological applications is the partial or total automation of the vehicle in these kind of situations. In the Intelligent Control Systems Laboratory of the Griffith University, in Australia, some autonomous vehicles, Cybercars, have the capability of performing an automatic route and dealing with basic intersection scenarios [5]. Another full autonomous vehicle driving application is that of the INRIA IMARA group in France. In this case and also using Cybercar vehicles, first steps in automatic intersection management are being carried out, allowing the cooperation of two of these cars in giving the way in intersections, using laser sensors and communications [6].A first simple case of use has been implemented.

In this paper we present the approach of the AUTOPIA Program of the Industrial Automation Institute of Spain for automatic driving in roadway intersection, based on GPS and wireless communications. We deal with the two simplest cases, in intersections in which the autonomous vehicle is circulating on a non-priority lane. These two cases of use are: the situation where a car is stopped in a priority lane and the autonomous vehicle circulates through the non-priority one and the situation where both cars are circulating in collision trajectory, with the autonomous going along the non-priority lane. Depending on its speed and position and the speed and position of the vehicle circulating over the priority lane, the autonomous driving system decides whether to stop or to continue the route. Some real experiments have been executed showing the performance of the system.

2 Vehicle Instrumentation

The AUTOPIA autonomous vehicle used in this work is a Citroën Berlingo van whose actuators have been automated. The throttle is managed directly by sending electronic analog signals to the gas control unit of the car. Brake is controlled using a DC motor engaged to the pedal through a pulley and commanded by a power/control card. Both elements are managed from an onboard computer where the control system resides that uses the AUTOPIA automatic driving architecture.

The sensorial input for automatic driving is provided by GPS and wireless communications. The automatic driving system is based on fuzzy logic and is able to determine

whether the vehicle has to stop and give the way to the priority one or to continue the route because the other vehicle is far enough to cross the intersection with safety.

3 AUTOPIA Automatic Driving Architecture

When designing an architecture that emulates human driving, we have to look at how humans organize the driving task and what operations they perform.

According to psychologists, human driving can be divided into three activity levels, depending on the attention, resources and perception that are applied. These are the strategic, tactical and control levels [7]. The strategic level includes planning, for example, choice of the best route to reach a destination. The tactical level comprises the execution of complex maneuvers like stopping, overtaking, giving way, etc. Finally, the control level refers to basic actions to keep the car on the right trajectory: moving the steering wheel, pressing the throttle or brake. These levels are ranked in descending order of complexity. This implies that the higher the complexity the more reasoning is needed and the less reactive the system is.

A control system based on human behavior that will support automated operation has to be built around an architecture paradigm. In our case we have chosen Michon [7] model, implemented as a hierarchical architecture, capable of supporting automatic driving and that can be upgraded to deal with other maneuvers that conform to human driving scheme. In our case the strategic planning stage has been taken over by manual user route selection. Then, our architecture is divided into six elements as shown in Fig. 1.

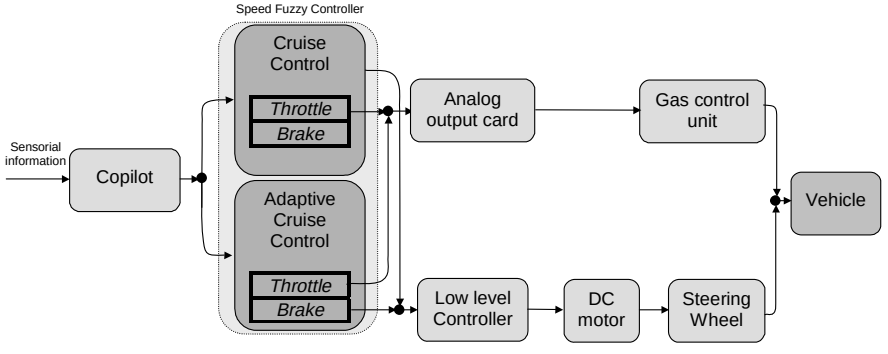


Fig. 1. Automatic speed control architecture

The first module is named copilot and emulates the tactical layer of human driving. It is a decision-making module whose mission resembles the job of a rally driving copilot. It tells the driver when the vehicle is entering a bend or a straight part of the route, when to increase or decrease target speed or when it is necessary to yield, controlling the sequence of operations to be carried out. Usually, the copilot manages the target speed with which the autonomous vehicle has to circulate though a segment of the road. It has also to select whether this speed control is simple (cruise control) or it is necessary to adapt the speed in order to keep a safety distance from a precedent

slower vehicle. In the case of intersection management, copilot aim is to manage the target speed of the vehicle, stopping or reducing its speed in the situations where it is necessary to give the way to another car that circulates in the other road of the intersection. Then it chooses between two kinds of speed behavior controllers: CC and ACC. These controllers represent the control layer of human driving and are modeled using fuzzy logic. This technique applies the knowledge of an expert operator, in this case a human driver, to control the equipment [8]. Another advantage is that complex mathematical models [9] are not needed to manage the equipment. This is a very useful feature where hard nonlinear systems, like vehicle throttle and brake control, are concerned. In other words, by applying fuzzy logic to control the speed of a car, we are modeling driver behavior and not the vehicle itself.

The throttle actuator consists of two additional modules: an analog output card that generates a proportional signal of the throttle fuzzy controllers output and the car gas control unit that selects the power effected by the motor according to this signal.

The brake management is somehow different. The third architecture module is the low-level controller. Its mission is to receive the target turning angle from the active fuzzy controller and to generate the appropriate control signals for the motor to move the brake. A PID, tuned to manage the DC motor and attached to the brake pedal, forms this low-level controller.

The fourth, fifth and sixth architecture modules are formed by the actual DC motor engaged by a pulley to the brake pedal.

4 Cases of Use

In this paper we consider two cases of use of the intersection management controller. The first scenario is the situation where the autonomous vehicle circulates through a non-priority lane towards the intersection while a second vehicle is stopped in the middle of the intersection in the priority lane. The second scenario is more complex. In this, a both cars circulate in a colliding trajectory and the automatic nonpriority one has to stop in order to yield the priority one.

4.1 First Case of Use

In figure 2 the configuration of this case of use,, the first scenario that must be solved by our automatic intersection situation manager is shown. The gray car represents the autonomous vehicle and the white one represents a car stopped in the center of an intersection. This is, for example, the case of a car that wants to turn to the left in the intersection or traffic congestion.

The automatic driving system controls the speed of the car, using digital cartography as reference. The GPS position also appears in the cartography of the intersection and the coordinates and the width of the cutting road. This information is used to reference the position of other vehicles from our route.

Once known the ego-position and the cartography of the involved roads, we can define a “collision area” as the portion of road, on the intersection, where a car on the priority lane can represents an obstacle in the route. It is, in this case, the piece of road where both lanes are overlapped.

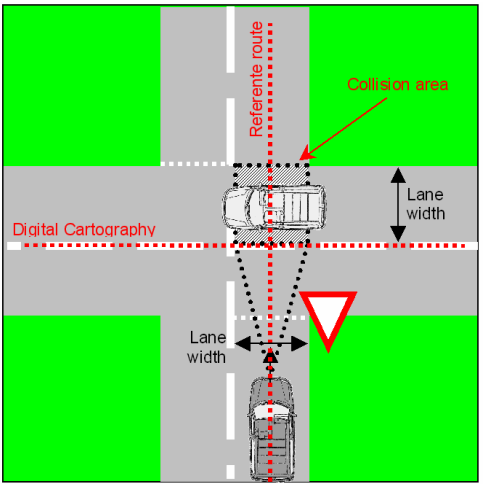


Fig. 2. Graphical representation of the first case of use of the automatic intersection management system

Once the collision area is located, the vehicle decides whether to stop through the environment information received from the WLAN: the GPS position in real rime and a timestamp for synchronizing the message reception.

The algorithm installed in the Copilot control module is in this case very simple: if the GPS position of the priority car is into the collision area and its speed is 0 km/h then stop at a safety distance. Else, continue route.

This system has been implemented and installed in our autonomous vehicle, executing a set of real experiments as the one shown in figure 3. In this case, there is a car stopped in the middle of the right lane, direction to the east, of the horizontal road (priority), manually driven and represented in the figure with a black triangle.

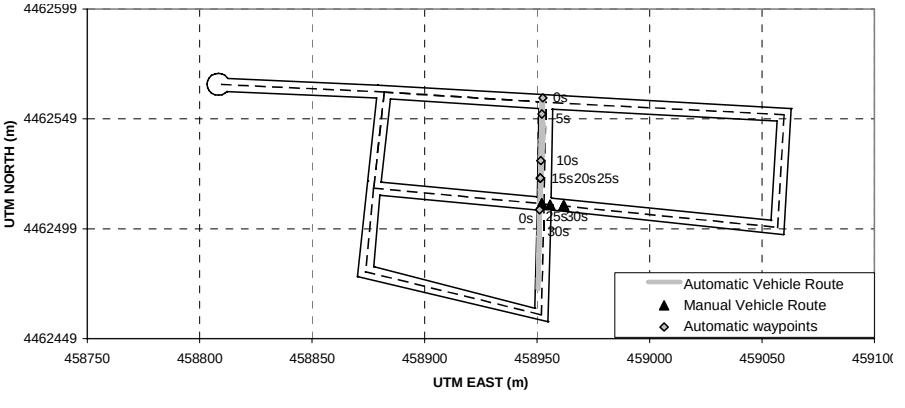


Fig. 3. Experiment of automatic intersection management. First case of use.

Autonomous vehicle trajectory is represented by a gray line and its temporal positions with a rhomboid. It starts its route in colliding trajectory, south direction, through the left lane of the road at 15 km/h. It receives the position of the other car by the wireless LAN environment and placer it in the collision area. When the autonomous vehicle is near enough the intersection (about second 15), the algorithm considers that it is at a safety distance and stops the car. At second 25, the stopped car starts moving and, when it departs from the collision area, the autonomous vehicle control system detects it and continue the route and cross the intersection without more interruptions.

4.2 Second Case of Use

Once solved the simplest case of use, we extend it to solve a more complex situation too. In this case, the intersection management system for autonomous vehicles has to deal with this situation when other vehicle approaches the crossroad in colliding trajectory (figure 4), circulating through a priority road (horizontal).

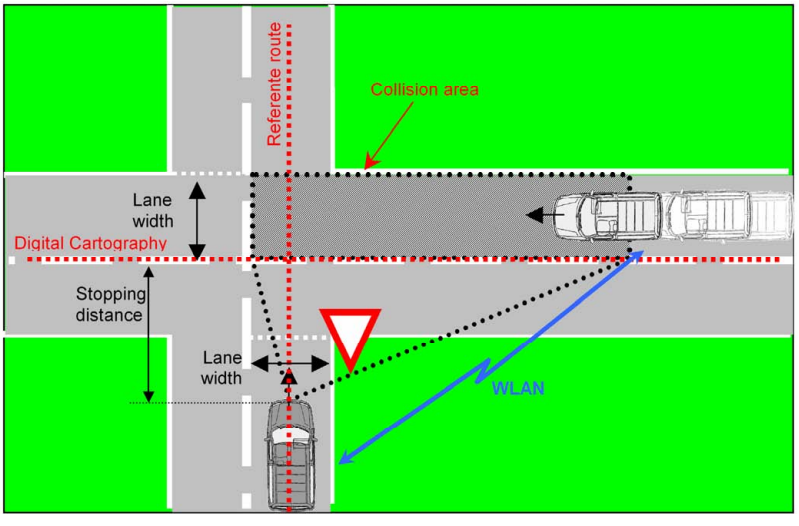


Fig. 4. Graphical representation of the second case of use of the automatic intersection management system

In this case, the gray vehicle is the equipped one that performs a guidance based on GPS. It also considers the cartography of the influence zone, detecting an intersection and a road that cuts its trajectory. Now, we redefine the collision area as the piece of the circulating lane of the priority road that start in the overlapping section up to a preset distance.

The control system, also knows the position of the other vehicle transmitted through the WLAN. Then, the yield algorithm is: *IF* the speed of the priority vehicle is not 0 (vehicle not stopped,) *AND* it circulates in colliding route *AND* it is in the collision area, *THEN* stop at a safety distance. *ELSE*, continue route.

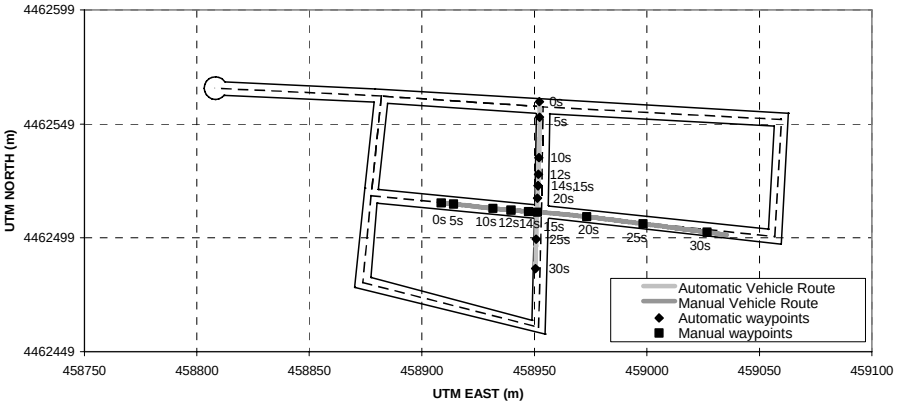


Fig. 5. Experiment of automatic intersection management. Second case of use.

Some experiments have also been carried out in order to demonstrate the feasibility of this system. In figure 5, a schema of the Industrial Automation Institute private testbed circuit is shown, considering the central horizontal road as priority and the central vertical as non priority. Manually driving vehicle circulates through the right lane of the priority road in direction to the East and its trajectory is represented by a dark gray line. Autonomous vehicle circulates through the non priority road, towards South, and its trajectory is represented with a light gray line. Rhomboidal an squared points represent the positions of both vehicles at the same instants from the beginning of the experiment. Both cars circulate about 15 km/h and they follow a collision trajectory.

The automatic intersection management system installed in the vehicle that circulates in the vertical road detects the other vehicle from the beginning of the experiment. It is circulating in a collision route, approaching to the center of the intersection. However, the vehicle is not in the collision area so no action is taken by the equipped car. From second 10, priory car enters in the collision area and the nonpriority one control system begin to take action, reducing speed until the car stops at second 14. From second 12 to 16, the priority the priority car is in the collision area and the equipped one is stopped at a safety distance. From second 16, the priority car surpasses the intersection; now, the way is free and the non priority can continue its route normally.

5 Conclusions

Two case of use for automating the intersection maneuver with ongoing traffic has been studied, implemented and tested. With this equipment, It is feasible to add to autonomous vehicles the capability of automatic intersection management. The required data to achieve this maneuver is: real time GPS position of the vehicles on the road, speed of the vehicles of the road, a digital cartography of the driving route, GPS timestamps for message synchronization.

In order to continue this work, our aim is to extend the behavior of the control system in yielding maneuvers. In these experiments, only stopping is a considered maneuver in order to respond a yielding. As future work we consider to reduce the circulation speed, optimizing the road occupancy and avoiding time looses in the trajectory following.

Acknowledgements. We want to acknowledge European Project Cybercars-2 (UE-STREP 28062) in whose framework is included this research.

References

1. Chan, C.: Characterization of Driving Behaviors Based on Field Observation of Intersection Left-Turn Across-Path Scenarios. IEEE T. on ITS 7(3), 322–331 (2006)
2. Misener, J.A., Ragland, D., Wang, X., Cohn, T.E.: Toward a Smarter, Safer Intersection: Application of Intelligent Countermeasures to the Intersection Crash Problem. In: Proc. of the ITS World Congress 2003, Madrid, Spain (November 2003)
3. Funderburg, K.A.: Update on Intelligent Vehicles and Intersections, US DoT Public Roads, vol. 67(4) (January/February 2004)
4. Fuerstenberg, K.Ch.: New European Approach for Intersection Safety – The EC-Project INTERSAFE, ITS Europe (2005)
5. Kolodko, J., Vlacic, L.: Cooperative Autonomous Driving at the Intelligent Systems Laboratory. IEEE Intelligent Systems 3, 8–11 (2006)
6. Bouraoui, L., Petti, S., Laouiti, A., Fraichard, T., Parent, M.: Cybercar cooperation for safe intersections. In: Proc. of the IEEE ITSC 2006, Toronto, Canada, pp. 456–461. IEEE Computer Society Press, Los Alamitos (2006)
7. Michon, J.: A Critical view of Driver Behavior Models: What do we Know, What Should we do? In: Evans, L., Schwing, R. (eds.) Human Behavior and Traffic Safety, Plenum, New York (1985)
8. Sugeno, M., Nishida, M.: Fuzzy Control of a Model Car. Fuzzy Sets and Systems 16, 103–113 (1985)
9. Mendel, J.M.: Fuzzy Logic Systems for Engineering: A Tutorial. Proceedings of the IEEE 83(3) (March 1995)

Free Space in Front of an Autonomous Guided Vehicle in Inner-City Conditions

Nicolas Simond

IMARA project-team - INRIA Rocquencourt
Domaine de Voluceau BP 105
78 153 Le Chesnay Cedex France
Nicolas.Simond@inria.fr

Abstract. This paper address the perception of the driving environment of an Autonomous Guided Vehicle from a vision system. The trajectory planning for an Autonomous Guided Vehicle requires to robustly segment the free space around the vehicle to manage an obstacle avoidance task. The urban environment still remains as the most complex environment due to the variety of dynamic elements which animate the scenes. The feature-based approach we detail in this article consists first on the segmentation of the ground surface to reconstruct finally a 2D polar map of the obstacles edges.

1 Situation

1.1 Related Works

Cybernetic Transport Systems have been developed for years to both enlarge the inner-city mass transport offer in a close future and reduce the traffic congestions. The navigation of a fully autonomous vehicle in our urban-like areas requires first we succeed to manage simultaneously the tasks of accurate localization, obstacle avoidance, trajectory planing and control. In this article, we focus on the extraction of the free-space in front of a vehicle with an embedded sterero-rig.

Most of authors use systems based on range-finders based on radars, Laser or Lidar to process task of obstacle detection. But this type of sensor remain expensive and their data generally requires the fusion with a vision system to segment the complex scenes. Therefore, many authors have been proposing for some years vision-based solutions.

Methods using a mono camera are generally based on a-priory knowledge of the scene and particularly on the type of obstacles. These methods are dedicated to detect either vehicles according their vertical edges, shadows, symmetry, lights [12] or pedestrians and crowd according to human shape and behaviour constraints. Some original methods track sets of features points [8] or edges [7] on obstacles because their relative motion is different from the camera motion. In a general way, the segmentation of the elements of a scene can be performed with a dense estimation of the optical flow [11] and the introduction of colour, texture constraints [3].

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1081–1088, 2007.
© Springer-Verlag Berlin Heidelberg 2007

The stereo-vision allows by triangulation the estimation of the depth of homologous pixels matched in a pair of images if the stereo-rig is calibrated. When the images are rectified, the computation of the V-disparity, an original method developed by Labayrade in [5], allows with real-time constraints the segmentation of obstacles whatever the longitudinal profile of the road and the discrimination between vehicles and trucks. Many authors perform the method with introducing U-disparity [4], infrared sensors to detect pedestrians [1] or multi-baseline sensors for off-road environment [2].

1.2 Motivations

The method we present in this paper is dedicated to Autonomous Guided Vehicles where the estimation of the free-space in front of the mobile robot is more crucial than the obstacle detection. In other words, all the elements which are lying above the road surface represent an obstacle, whatever their shapes or elevations. Hence, the kerbs and the green strips which generally highlight the roads boundaries have to be segmented as obstacles if they have a non-null elevation.

This paper is organized as follows: in Section II, we propose a method to compute easily a bird eye view of the road plane. The identification of the edges lying above the considered plane is presented in Section III. The interest of the representation of the obstacles detection stage with an IPM image is presented in Section IV while we conclude and propose some possible research directions in Section V.

2 Inverse Perspective Mapping

2.1 Principle

According to our previous works and the restricted field of view when a vision system is embedded in an urban vehicle, we have implemented a simple methodology to compute an *I*nverse Perspective Mapping (IPM) image [6]. The IPM transformation consist on modifying the angle of view under which a scene is acquired to remove the perspective effect. In a practical way, the IPM transformation consist to set the angles of rotation (mainly the tilt one) to values such as the normal to the plane $\mathbf{n}$ and the focal axis of the virtual camera $\mathbf{z_v}$ are opposite: $\mathbf{n}.\mathbf{z_v} = -1$, according to the Fig.1. The virtual camera follows the vehicle path but has a constant ordinate z_v along its $\mathbf{z}$ axis, whatever the slope of the road s.

The method only requires the extraction of two coplanar parallel lines. Due to the perspective effect, the 3D lines which are parallel have projections in image which converge to a Vanishing Point. The IPM transformation also consist on determining the point of view of a virtual camera where the projections of the two lines appear parallel.

In case of structured road, the navigation lanes are usually highlighted with parallel painted markers. Hence, the extraction of two parallel lines along a

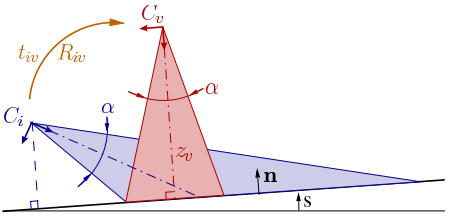


Fig. 1. Principle of the IPM transformation: the virtual camera $\mathcal{R}_v$ has a constant elevation z_v along the third $\mathbf{z_v}$ axis which is parallel to the normal to the plane $\mathbf{n}$ whatever the camera pose $\mathcal{R}_i$ and the road slope s while the aperture angle α is maintained.

straighted path is relatively easy. Many authors propose several models of the road projection that mainly differ on the free field of view, the presence of road markers or not, the number of degrees of liberty the road model assumes. We personally use the simplest model: the road borders are linearized at the foreground and converge to the *Dominant Vanishing Point* (DVP) $\mathbf{x}$ whatever the road curvature is.

The reader can retrieve in [10] the tracking process we have developed to segment the projection of the road in images. The complete methodology of the computation is detailed in [9]. We nevertheless propose a resume in the following. Due to the IPM transformation is equivalent to a change of view for a planar scene, it is obvious that this transformation is a homography. By definition, a homography is defined with four couples of matched features between two images of the same plane.

The methodology is based on three main assumptions at the image foreground:

1. the road is locally flat in the first meters in front of the vehicle,
2. the two projections of the road boundaries can be linearized as two lines which converge to the *D*ominant Vanishing Point (DVP) $\mathbf{x}$,
3. the road width is constant that induces the boundaries are locally parallel whatever the radius of curvature.

Hence, the intersections of the two road boundaries with two horizontal lines generate four corners $\mathbf{p}_{\pi\mathbf{k}}$ with $k \in \{1, 2, 3, 4\}$ which represent a trapezoid in original image. After the IPM transformation, the four points have to represent a parallelogram where the road boundaries have to be parallel. The troubles we also face are how to determine the two parallel lines and what are the locations of the four points in the IPM image ?

Lets now considering the horizontal lines $\mathbf{l_b}$ and $\mathbf{l_t}$ whose v-ordinates are the image height $v_{b(ottom)}$ and an other upper $v_{t(op)}$. If the road has a constant width and the elevation z_v of the virtual camera remains constant, the road representation in the IPM image $w_{\pi v0}$ is constant too whatever the evolution of the camera pose.

Furthermore, the aperture angle of the camera lens α is supposed to be known and constant. The IPM transformation can be resumed coarsely in a first

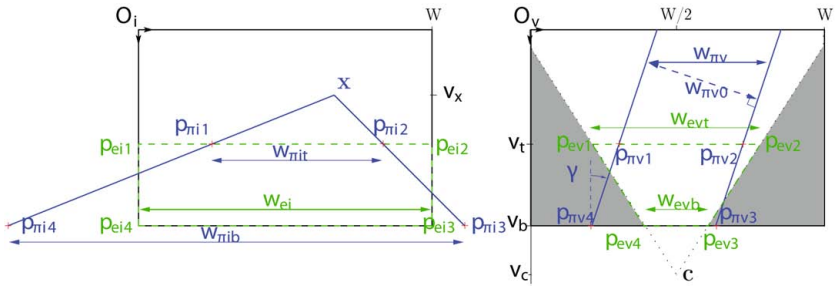


Fig. 2. Principle of the IPM transformation: the road boundaries converge to the DVP $\mathbf{x}$ in $\mathcal{R}_i$ and appear parallel in $\mathcal{R}_v$. The intersections of the image boundaries and the road borders with the two parallel lines $\mathbf{l_b}$ and $\mathbf{l_t}$ respectively generate the points $\{\mathbf{p_{e1}}, \mathbf{p_{e2}}, \mathbf{p_{e3}}, \mathbf{p_{e4}}\}$ and $\{\mathbf{p}_{\pi\mathbf{1}}, \mathbf{p}_{\pi\mathbf{2}}, \mathbf{p}_{\pi\mathbf{3}}, \mathbf{p}_{\pi\mathbf{4}}\}$.

approximation as a rotation of the camera framework around its $\mathbf{u}$ axis with an angle close to 90° when the camera axis is parallel to the road surface. That induces with a pinhole model for the camera the IPM image represents the original image with a cone whose aperture angle is α and the location of the cone summit $\mathbf{c}$ is invariant.

By definition, the homography $\mathsf{H_{vi}}$ which is the IPM transformation only check the cross-ratio consistency and the incidence. That implies the two quadruplets of points, form with the $\mathbf{p}_\pi$ and $\mathbf{p_e}$ points, along the $\mathbf{l_b}$ and $\mathbf{l_t}$ lines have invariant cross-ratio. Due to we assume these two lines project in the IPM image with the same horizontal direction, the IPM transformation is equivalent to an affine transformation, that induces the ratio and the relative location between the road width and the image width have to be checked:

$$\frac{w_{\pi it}}{w_{ei}} = \frac{w_{\pi v}}{w_{evt}} \quad \text{and} \quad \frac{w_{\pi ib}}{w_{ei}} = \frac{w_{\pi v}}{w_{evb}} \tag{1}$$

The estimation of the homography $\mathsf{H_{vi}}$ is also straight forward whatever the quadruplet of corners considered:

$$\begin{cases} \mathbf{p}_{\pi\mathbf{vk}} \simeq \mathsf{H_{vi}}.\mathbf{p}_{\pi\mathbf{ik}} \\ \mathbf{p_{evk}} \simeq \mathsf{H_{vi}}.\mathbf{p_{eik}} \end{cases} \text{ according to } \begin{cases} \mathbf{l_{b(v)}} \simeq \mathsf{H_{vi}^{-t}}.\mathbf{l_{b(i)}} \\ \mathbf{l_{t(v)}} \simeq \mathsf{H_{vi}^{-t}}.\mathbf{l_{t(i)}} \end{cases} \tag{2}$$

The u-ordinate of the cone summit is assumed at the middle of the image width whereas the v-ordinate v_c is computed a single time to fix the road width $w_{evb}/2 = (v_c - v_b)\tan(\alpha/2)$ such as $w_{\pi v0} = w_{\pi v}\cos(\gamma)$. To remove the parameter γ, we personally choose a frame where the focal axis of the camera is close to the road direction ($\gamma \sim 0°$).

If it is obvious that the bottom horizontal line project itself into its own location in the IPM image, it is not so trivial for the top line. We nevertheless suppose that the two horizontal lines are invariant with the IPM transformation. We also obtain one solution to the IPM transformation which has the great interest to be computed easily even if the camera pose and internal calibration are unknown.

The lonely drawback of this solution is the pixels of the IPM image have a rectangle shape. To counterbalance this imperfection, we had to determine the remaining parameter the v-ordinate of the top line in the IPM image. This operation requires the calibration of the internal parameters of the camera or in a practical way to observe with the camera a coplanar stripe whose the dimensions are known to estimate the scale factor along the v-axis of the IPM image.

A remaining parameter have to be identified: the v_t v-ordinate of the top line $\mathbf{l_t}$. This parameter highly depends on the DVP location when the Thales theorem is applied to the triangles $\{\mathbf{x}, \mathbf{p_{\pi i3}}, \mathbf{p_{\pi i4}}\}$ and $\{\mathbf{c}, \mathbf{p_{ev1}}, \mathbf{p_{ev2}}\}$ which are linked thanks to the IPM transformation. The global system is also constrained although the scale factor along the v-axis of the virtual camera is unknown:

$$v_t = v_c + v_x - v_b. \tag{3}$$

Due to we don't possess a ground truth, the calibration of the camera is not available. We hence suppose that the aperture angle α equal to 60° and we empirically fix the road width $w_{\pi 0}$ in the IPM image equal to 200 pixels. We present in Fig.3 the result of the IPM transformation with the preceding method. The (blue) road markers which border the navigation lane converge to the DVP in the original image and appear parallel in the IPM image. The intersection of the pencils of lines with the two (dashed green) horizontal lines provide the (red) corners we use to compute the IPM transformation: they allow the estimation of the camera orientation γ is true up to the unknown scale factor.

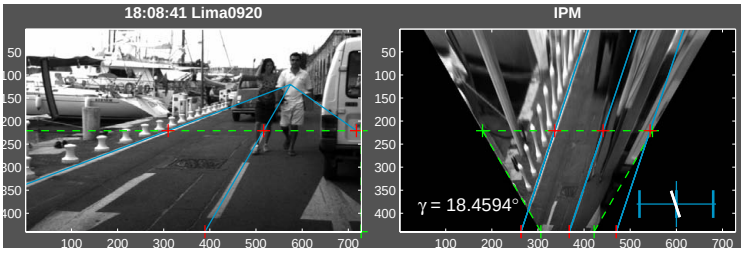


Fig. 3. The IPM transformation is computed thanks to the intersections of the first and the third blue medians of the road boundaries with the green dashed $\mathbf{l_b}$ and $\mathbf{l_t}$ lines. The second median is shown for convenience.

3 Identification of the Obstacles Edges

The accuracy of the homographies extracted from the Super-Homography allows us to detect all the elements which are above the considered plane. Thanks to the feature-based method we have developed cf. [10], we had the choice to detect the obstacles with points and/or lines. We nevertheless use edges because the obstacles have sometimes uniform textures.

We present in this section a two-stages method where the coplanar edges are first identified to be removed, then the remaining edges are projected with an IPM transformation to determine the regions free from obstacles. We also only focus on edges which have both extremities under the horizon line, assumed as the horizontal line which has the same v-ordinate as the DVP v_x.

3.1 Identification of Coplanar Edges with Homography Then Correlation

Although any couple of images copes with the following method, we present results extracted from a couple of stereo images. The discrimination between the road and obstacles is more difficult if the images are issued from two consecutive views of the same camera. At low speed, the relative speed of the scene elements are too close to reliably discriminate them, it is not possible to detect obstacles which have a speed similar to the camera speed and there exist no solution to identify coplanar elements which are obstructed in only one image.

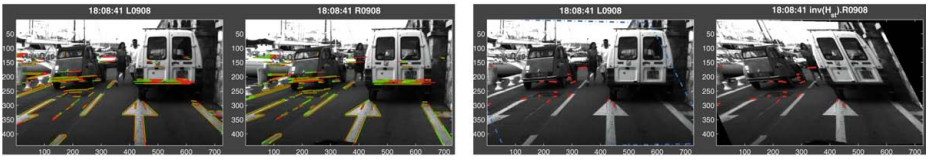


Fig. 4. Sequential search of the coplanar edges in the left (red) image thanks to the right (green) image and the stereo homography H_{st}. At left are shown the edges which are overlapping by retro-projection between the left and right images. At right, the edges have similar neighborhood in original and warped images.

The edges shown in Fig.4 *left*, issued from the left (red) and right (green) cameras, check the overlapping criterion. As expected, the majority of outliers (matched above the road plane) are horizontal edges whose the pan angle between the camera framework is not enough large to discriminate them with such a method whatever the camera motion. We actually obtain similar results when we consider the camera motion between two consecutive poses. The best way is certainly do not take account the horizontal edges in this procedure because the determination of their coplanarity is not reliable.

At this point, the majority of the coplanar edges are identified. The coplanar edges still remain because they don't fit precisely the contours they represent or they don't have homologous in the other view due to occlusion or illumination effects. The rejection of these edges requires one additional process to verify their planarity. We also compute the *S*um of Absolute Difference (SAD) image, as the difference between the warping of one image into the other according to the homography relative to the camera motion between the two poses, highlight the scene regions where the assumption (planarity, same relative motion) is checked thanks their gray levels are the darkest.

The discrimination between edges lying on or above the road with SAD image is not easy: due to the illumination conditions the areas close to the edges often appear clearest, whatever the quality of the estimation of homography. Therefore we prefer compute the normalized correlation in 3 pixels large region around each edges in the original and its homologous, the other image expressed in the original framework. The coplanar edges identified in Fig.4 *right* have either high correlation score or low meaning gray level with correlation score less high. This procedure allows the identification of coplanar edges which have no homologous in the other image.

4 Free-Space Estimation

The best way to locate the edges above the road is using an IPM transformation to produce a polar map of the obstacles edges. The region of the IPM image free from the remaining edges is free from obstacles. The segmentation of the obstacles still remains difficult but the reliability of the free-space is high.

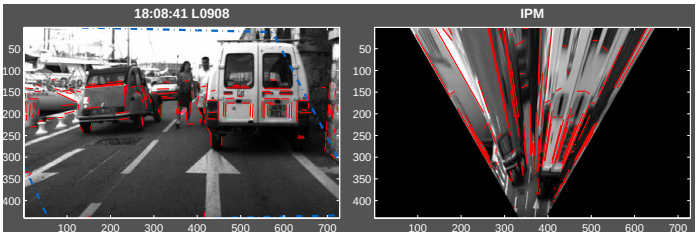


Fig. 5. The free-space in front of the camera is the lowest region of the IPM image under the obstacles edges

Opposite to most of the methods developed on the reconstruction of the 3D scene based on (V-)disparity, the proposed method doesn't make assumption on the obstacle elevation. The limitation of the method is the length of the edges which have to be greater than 10 pixels, that allows us to detect the majority of the white bollards which are lying on left road boundary of the Antibes sequence like in Fig.5. That is our goal for an implementation in an AGV.

5 Conclusion

We present in this paper a methodology to compute IPM image from planar scene where two parallel lines can be extracted. The computation of the Super-Homography allows the identification of the edges which are not lying on the considered plane. The IPM image of the remaining edges provides a polar map of the free-space in front of the stereo-rig which acquires the images.

The use of IPM on images of a stereo-rig drastically improves the correlation score of the coplanar features. The introduction of the IPM transformation in the Super-Homography computation could reduce the time processing of the identification of the coplanar features and the homography computation. The estimation of the free-space in front of the camera have to be improved to segment a convex region free from obstacles.

References

1. Bertozzi, M., Binelli, E., Broggi, A., Del Rose, M.: Stereo vision-based approaches for pedestrian detection. In: CVPR 2005. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Los Alamitos (2005)
2. Broggi, A., Caraffi, C., Porta, P.-P., Zani, P.: A single frame stereo vision system for reliable obstacle detection during darpa grand challenge 2005. In: ITSC 2006. IEEE Intelligent Transportation Systems Conference, IEEE Computer Society Press, Los Alamitos (2006)
3. Heisele, B.: Motion-based object detectin and tracking in color images sequences. In: ACCV 2000. Fourth Asian Conference on Computer Vision, January 8-11, Taipei, China, pp. 1028–1033 (2000)
4. Hu, Z., Lamosa, F., Uchimura, K.: A complete u-v-disparity study for stereovision based 3d driving environment analysis. In: 3DIM 2005. Fifth International Conference on 3-D Digital Imaging and Modeling, pp. 204–211. IEEE Computer Society Press, Washington, DC, USA (2005)
5. Labayrade, R., Aubert, D., Tarel, J.-P.: Real time obstacle detection in stereovision on non flat road geometry through "v-disparity" representation. In: IV 2002. IEEE Intelligent Vehicle Symposium, France, pp. 17–21 (June 2002)
6. Mallot, H.A., Blthoff, H.H., Little, J.J., Bohrer, S.: Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biological Cybernetics (1991)
7. Okada, R., Taniguchi, Y., Furukawa, K., Onoguch, K.: Obstacle detection using projective invariant and vanishing lines. In: ICCV 2003. 9th IEEE International Conference on Computer Vision, Nice, France, pp. 13–16. IEEE Computer Society Press, Los Alamitos (2003)
8. Rabie, T., Auda, G., El-Rabbany, A., Shalaby, A., Abdulhai, B.: Active-vision based traffic surveillance and control. In: VI 2001. Vision Interface Annual Conference, June 7-9, Toronto, Canada (2001)
9. Simond, N.: Free-space from ipm and super-homography. In: IROS 2007. IEEE RSJ/International conference on Intelligent Robot and System, San Diego, CA, USA (October 2007)
10. Simond, N., Laurgeau, C.: Vehicle trajectory from an uncalibrated stereorig with super-homography. In: IROS 2006. IEEE RSJ/International conference on Intelligent Robot and System, Beijing, China, IEEE Computer Society Press, Los Alamitos (2006)
11. Stein, G.P., Mano, O., Shashua, A.: A robust method for computing vehicle egomotion. In: IV 2000. IEEE Intelligent Vehicle Symposium, Piscataway, NJ, USA, October3-5, pp. 362–368 (2000)
12. Steux, B., Laurgeau, C., Salesse, L., Wautier, D.: Fade: a vehicle detection and tracking system featuring monocular vision and radar data fusion. In: IV 2002. IEEE Intelligent Vehicle Symposium, IEEE Computer Society Press, Los Alamitos (2002)

Towards a Robust Vision-Based Obstacle Perception with Classifier Fusion in Cybercars

Luciano Oliveira, Gonçalo Monteiro, Paulo Peixoto, and Urbano Nunes

Institute of Systems and Robotics
Department of Electrical and Computer Engineering
University of Coimbra, 3030-290, Coimbra, Portugal
{lreboucas, ggmonteiro, peixoto, urbano}@isr.uc.pt

Abstract. Several single classifiers have been proposed to recognize objects in images. Since this approach has restrictions when applied in certain situations, one has suggested some methods to combine the outcomes of classifiers in order to increase overall classification accuracy. In this sense, we propose an effective method for a frame-by-frame classification task, in order to obtain a trade-off between false alarm decrease and true positive detection rate increase. The strategy relies on the use of a Class Set Reduction method, using a Mamdani fuzzy system, and it is applied to recognize pedestrians and vehicles in typical cybercar scenarios. The proposed system brings twofold contributions: i) overperformance with respect to the component classifiers and ii) expansibility to include other types of classifiers and object classes. The final results have shown the effectiveness of the system[1].

1 Introduction

Artificial vision systems applied to vehicles play an important role, sensing the surrounding in a dense way and identifying concerning objects from image. The main framework of most of these systems is usually formed by four general modules: acquisition, pre-processing, feature extraction, recognition and interpretation. Although all modules should achieve a strong robustness in order to obtain a consistent system, one may say that the kernel of these systems lies on the feature extraction and recognition modules.

Throughout the years, many works have been carried out in order to extract and to classify relevant information using single classifiers [1], [2]. Since this approach seems to have constraints because the particular nature of applications, some methods have been suggested toward to combine the outcomes of different classifiers in order to increase the overall performance of these systems [3]-[5].

[1] This work is supported in part by Fundação para a Ciência e Tecnologia (FCT) de Portugal, under Grant POSC/EEA-SRI/58279/2004, and by CyberC3 project (European Asia IT&C Programme). Luciano Oliveira is supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), programme of Ministry of Education of Brazil, scholarship n^o BEX 4000-5-6.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1089–1096, 2007.
© Springer-Verlag Berlin Heidelberg 2007

A comprehensive study of the existing techniques to fuse data from component classifiers is found in [6].

In [7], Monteiro *et al* have introduced a complete framework to recognize pedestrians and vehicles (see Fig. 1). This is composed by four subsystems: an Adaboost vision-based system, a lidar-based system, a coordinate transformation system, and a tracking and final classification system. Though these four subsystem work together, here we are focusing our attention in the vision-based system and is there where we applied our novel architecture. This new vision system is now based on a classifier fusion, using two classifiers: an Adaboost [1], and a linear SVM [8]; with different feature spaces: Haar-like features [1] and dense Histograms of Oriented Gradients (HOG) [9], respectively. To carry out the fusion, a two tier Mamdani fuzzy system is utilized, with the underlying goal of obtaining a trade-off between false alarm decrease and true positive detection rate increase, using a Class Set Reduction (CSR) strategy. The objective of this module is to decide whether the outputs should be maintained or not according to the neighborhood intersection of the two class labels and their posterior probabilities (score confidence of the classifiers). The system provides a high flexibility not only for adding new object classes, but also for allowing the adjustment of the fuzzy parameters to tune the fusion task appropriately. Nevertheless, the use of a global feature-based classifier (Haar-like feature / Adaboost) and a local feature-based classifier (dense HOG / SVM) brings additional advantages to cope with the problem of illumination sensitivity and partial occlusion.

Contents. In §2, some related works are briefly resumed. §3 gives some details of our sensing system. In §4, we explain our method, as well as present the structure of each used classifier. §5 details our validation methods and final results, using Detection Error Tradeoff (DET) curves. Finally, §6 draws some conclusions and future work.

2 Related Works

A combination of multiple classifiers using fuzzy templates in order to aggregate the output labels according to their posterior probability is proposed in [3]. It only works with global feature-based classifiers. More recent works are presented in [4] and [5]. The first one introduces two methods to combine local and global features: a "stacking" ensemble technique, which performs in the output labels of the classifiers, combining their output vectors and classifying these vectors in a meta-classifier; and a Hierarchical Classification System (HCS), which first uses global feature-based classifiers and, then, tries to validate this output with a local feature classification. In [5], a similar approach to HCS is discussed. The main difference between both of them is the fact that the latter one computes the posterior probability to infer whether to apply local feature computation or not. These three works discussed in this section have some similarities with our work, differing in two points: the work that explores an adaptive way of integrating classifiers just uses global feature classification and, thus, fail in treating

occlusion and in taking advantage of other strengths encountered in local feature-based classifiers; those that use the combination of global and local features, tend to perform it in a sequential way, first applying a global feature classification, and hence discard some beneficial characteristics of local feature-based classifier in the first stage of classification.

3 Architecture of the Sensing System

Our sensing system was first presented in [7]. It is continuously being developed and modified in order to achieve a higher performance system. It is composed by four subsystems: a vision-based system (responsible to recognize pedestrians and vehicles using an Adaboost classifier), a lidar-based system (not only it recognizes pedestrians and vehicles, but also provides a Region of Interest (ROI), working as a focus of attention to speed up the vision classification), a coordinate transformation system (a transformation matrix, achieved in a calibration task, responsible to transform world coordinates into image coordinates) and the tracking and final classification system (providing the dynamic features of each object being tracked). The architecture of the sensing system is depicted in Fig. 1.

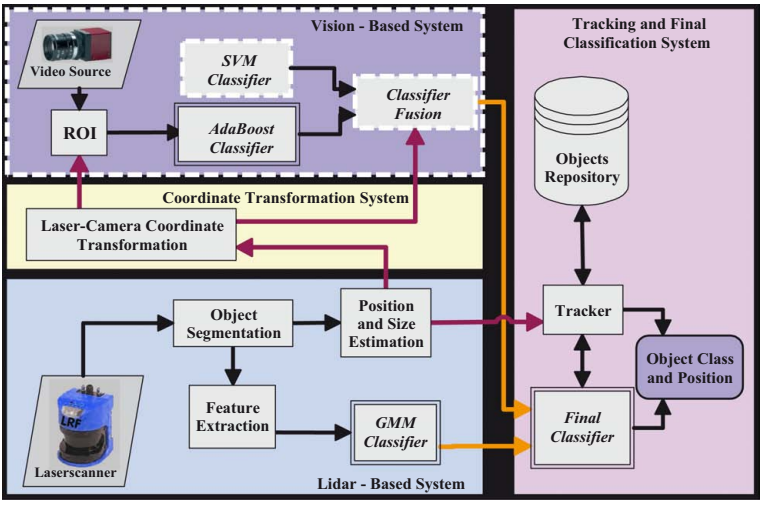


Fig. 1. Overall architecture using laser and vision information for object detection, classification and tracking

4 Proposed Vision-Based System

A novel vision-based system, integrated in the overall architecture (white dashed lines in Fig. 1), is proposed here, mainly as an attempt to decrease the illumination sensitivity and to include a partial object occlusion characteristic. The

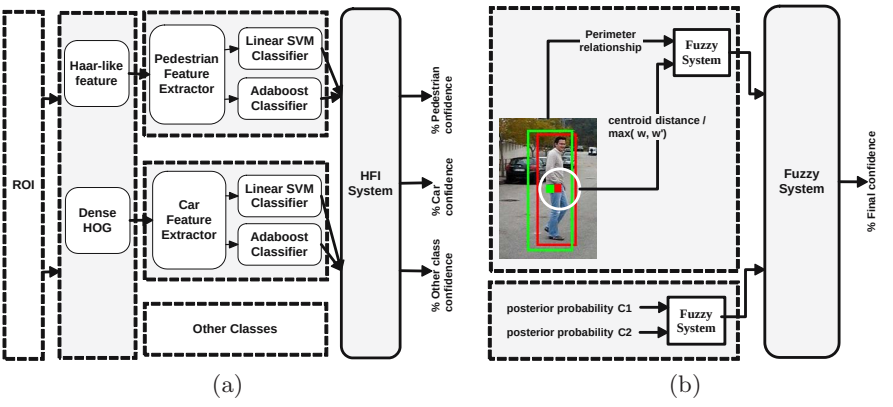


Fig. 2. In (a), the overall architecture of the proposed system applied in vision-based system in Fig. 1. In (b), the internal structure of the HFI system: the first tier is responsible to achieve confidences of the classifiers according to posterior probabilities and geometry; in the last tier, all confidences are fused in another fuzzy system to give the final confidence. w and $w^{'}$ represent the width of the outputs of each classifiers; $C1$ and $C2$ are Adaboost and Linear SVM classifiers.

general architecture of the proposed vision-based system, which has been denominated Hierarchical Fuzzy Integration (HFI), is illustrated in Fig. 2(a). The internal structure of the HFI system is depicted in Fig. 2(b). After finding the percentage of the confidence with respect to the neighborhood intersection of the labels and their posterior probabilities, for each detected object, a threshold is applied to evaluate the final result according to a predefined observed behavior of the classifiers. The architecture is expansible to include other types of classifiers and object classes. For completeness, the structure of each component classifier will be described in the following section.

4.1 Component Classifiers

Haar-like feature / Adaboost. An Adaboost has been used to classify a set of haar-like features, which are represented by templates [1]. The paradigm of this classification method resides in the computation of these features which are combined to describe an object. To accomplish this, a set of so called weak classifiers is assigned to select every feature that best separate the foreground and background (hence, it is considered a global feature-based classifier). For each feature, a weak classifier determines an "optimal" threshold. Each weak classifier $h_j(x)$ (where x is a $m \times n$ pixel sub-window of an image) consists of a feature f_j, a threshold Θ_j and a parity p_j which indicates the direction of the inequality sign, according to:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j < p_j \Theta_j \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Each of $h_j(x)$ classifier reacts to a haar-like feature. The final classifier $F(x_i) = \sum_{k=1}^{N} \alpha_k h_j(x_i)$ is composed by all weak classifiers, where x_i is the ith element of the input vector, α_k is a weight function of each weak classifier $h_j(x_i)$ and N is the number of weak classifiers. An object is only considered if it goes through all stages of weak classification.

Dense HOG / SVM. Histogram of Oriented Gradient descriptor has reached a maturity version, for object matching, in [10] and it was further modified by [9], in a pedestrian localization system in images. This new implementation is formed by a twofold process, illustrated in the following algorithm.

Algorithm 1. Training and Learning Stages of HOG / SVM

Training stage
- Extract feature vectors of normalized positive samples and random negative samples
- Normalize vectors
- Run classifier against negative samples, thouroughly
- Retrain hard examples
- Create final linear SVM model

Learning Stage
- Scan images at different location and scale
- Predict classes with linear SVM model
- Fuse multiple detections with meanshift clustering algorithm and
- Give object bounding boxes in image

Firstly, a set of HOG features is extracted densely across the image. After feature normalizzation, feature vectors are passed to a linear SVM in order to obtain a model of support vectors. In case of negative features, which are extracted randomly across non-object images, the model is retrained with an augmented hard examples to improve performance. The classification stage scans the input image over several orientations and scale, and after the detection of all objects, a mean shift algorithm is performed to cluster the concerning detection windows and to detect the final localization of the objects in the image. For all stages, several parameters may be used and there exists a set of default optimal values [9].

4.2 Architecture of the Fuzzy Systems

The architecture of all fuzzy systems used is illustrated in Table 1. The universes of discourse of all variables are in the interval [0, 100], representing rate or confidence values. In the first tier (see Fig. 2), the inputs are the overlap area of the bounding boxes (of each classifier) and the posterior probabilities. Each classifier gives its confidence score by obtaining the number of window detection that surrounds each object and determining the minimum-maximum interval, considering a test set. The defuzzification method used was Mean of Maxima [11].

Table 1. Characteristics of the fuzzy systems

Fuzzy system	Variables	Fuzzy sets
Class label	Distance ratio	{Near, Medium, Far}
	Perimeter ratio	{Little, Medium, Big}
	Intersection confidence	{Low, Medium, High}
Posterior prob.	Adaboost post. prob.	{Low, Medium, High}
	HOG / SVM post. prob.	{Low, Medium, High}
	Post. prob. confidence	{Low, Medium, High}
Final	Intersection confidence	{Low, Medium, High}
	Post. prob. confidence	{Low, Medium, High}
	Final confidence	{Low, Medium, High}

5 Experimental Results

5.1 Methodology

Experiments were conducted in order to evaluate the system. Two training data sets have been used: for pedestrians, the INRIA dataset, proposed by [9], and, for cars, the CALTECH dataset [12]. The INRIA dataset is composed by 2416 pedestrian images and 12180 background images, for the training phase and 997 pedestrian images of realistic scenes, for the test phase. The CALTECH dataset is composed by 3698 car images and 13690 background images, for the training phase and 337 car images, for the test phase.

5.2 Validation

After gathering the datasets involved to train and validate the system, we have trained and run both component classifiers against the test set to evaluate their performance. This was made by plotting DET curves of each classifier and final

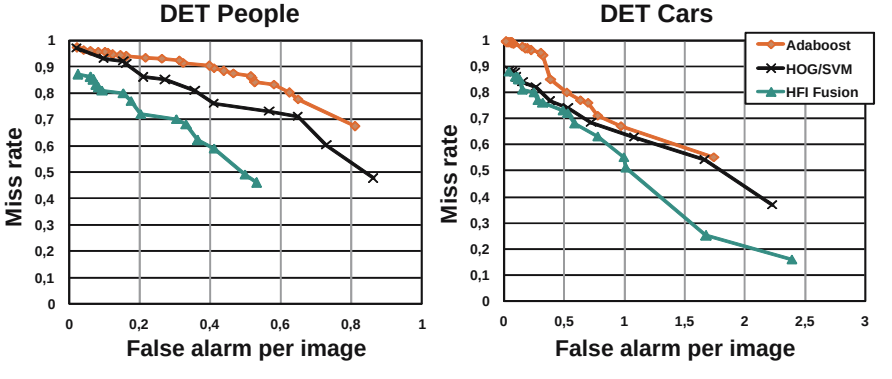


Fig. 3. DET curves. The proposed method outperforms the component classifier.



Fig. 4. Result samples: left picture shows the result of the Adaboost classifier; central picture shows the result of the HOG/SVM classifier and the right picture shows the fusion result

fusion, according to Fig. 3, where the lower is the curve, the better the performance of the classifier. We used the annotations provided by the two datasets to validate the system and to plot the DET curves. In order to assign a true detection, we used the following overlap criterion, where OA stands for Overlapped Area (must be greater or equal to 0.5, to a given class c be correctly considered an object in the image), A_{gt} is the area of the bounding box assigned for the annotation (ground truth) and the A_c is the area of the bounding box assigned for the classifier. In Fig. 4, some results of the classification fusion, in a street environment, are shown.

$$OA = \frac{A_{gt} \cap A_c}{A_{gt} \cup A_c} \geq 0.5. \tag{2}$$

6 Conclusions and Future Work

A fuzzy CSR method to combine the outputs of local and global feature-based classifiers has been presented in this paper. The proposed system combine the strengths of two well-established classifiers to incorporate invariance concerning to illumination and partial occlusion (coming from the local feature-based classifier). The idea of including the posterior probability of the classifiers overcame the problem of discarding labels that do not intersect each other in certain circumstances. The proposed fusion system has overperformanced its components classifiers, although the final tradeoff between false positive and miss detection rate has not reached a desirable value. These conclusions lead us to better tune the classifiers and to investigate other methods of classification. We are now investigating other ways of combining global and local feature-based classifiers, including an extended evaluation of other classifiers and ways of combination. Since we use a laserscanner in our complete framework, the idea of using more classifiers should not drastically affect the computational time of the overall system, as the assigned classification area of the image comes from the laserscanner ROI.

References

1. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 511–518. IEEE Computer Society Press, Los Alamitos (2001)
2. Sotelo, M., Parra, I., Naranjo, E.: Pedestrian Detection using SVM and Multi-feature Combination. In: IEEE Intelligent Transportation Systems Conference (ITSC), pp. 103–108. IEEE Computer Society Press, Los Alamitos (2006)
3. Kuncheva, L.I., Bezdek, J.C., Sutton, M.A.: On Combining Multiple Classifiers by Fuzzy Templates. In: NAFIPS Conference of EDS, pp. 193–197 (1998)
4. Lisin, D.A., Mattar, M.A., Blaschko, M.B., Learned-Miller, E.G., Benfield, M.C.: Combining Local and Global Image Features for Object Class Recognition. In: International Conference on Computer Vision and Pattern Recognition, pp. 47–55 (2005)
5. Murphy, K., Torralba, A., Eaton, D., Freeman, W.T.: Object Detection and Localization Using Local and Global Features. LNCS, pp. 393–413 (2005)
6. Ruta, D., Gabrys, B.: An Overview of Classifier Fusion Methods. Computing and Information Systems, 1–10 (2000)
7. Monteiro, G., Premebida, C., Peixoto, P., Nunes, U.: Tracking and Classification of Dynamic Obstacles Using Laser Range Finder and Vision. In: Workshop on Safe Navigation in Open and Dynamic Environments, held at the IEEE International Conference on Intelligent Robots and Systems (IROS), pp. 213–219. IEEE Computer Society Press, Los Alamitos (2006)
8. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
9. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: International Conference on Computer Vision and Pattern Recognition, pp. 886–893 (2005)
10. Mutch, J., Lowe, D.G.: Multiclass Object Recognition with Sparse, Localized Features. In: IEEE International Conference on Computer Vision and Pattern Recognition, pp. 11–18. IEEE Computer Society Press, Los Alamitos (2006)
11. Tsoukalas, L.H., Uhrig, R.: Fuzzy and Neural Approaches in Engineering, p. 165. Wiley-Interscience Publication, Chichester (1996)
12. Fei-Fei, L., Fergus, R., Perona, P.: One-Shot Learning of Object Categories. IEEE Transactions Pattern Recognition and Machine Intelligence, 594–611 (2004)

Using Self-adapting Navigation Data for Intelligent, Personalized Vehicle Guidance

Michal Veselý and Herwig Mayr

Research Center Hagenberg
Department of Software Engineering
Upper Austria University of Applied Sciences
Softwarepark 11, A-4232 Hagenberg, Austria
{michal.vesely, herwig.mayr}@fh-hagenberg.at

Abstract. Even though computer-based navigation systems, being developed since the late 1980s, are still among the key topics worked on in information technology and the road network changes circa 15% per year, there are no significant possibilities for *individual* update and adjustment of the digital map data.

In this paper, we present the concepts of our software system under development, which aims at the automatic construction and extension of digital maps, thus enabling continuous improvement of their quality as well as allowing for advanced, intelligent, and personalizable navigation, depending on user behavior and preference. This approach comprises an appropriate processing of the raw data, reliable graph optimization and merging techniques, and finally, suitable data exchange interfaces.

Keywords: navigation personalization, digital maps, incremental update, ADAS, GPS.

1 The Need for Up-to-date Digital Map Data

All common navigation systems rely on a similar architecture: using a road database in the manufacturer's proprietary format, a route calculation module, a positioning module processing the Global Positioning System (GPS) data (see [6] for details), a map matching module (see [10]), and a guidance module. Vehicle navigation systems can calculate a route along the road network available in the database from the current car location to the destination point, and guide the driver via acoustic and visual instructions.

Even though navigation systems have become more reliable and affordable during the last few years, the navigation *data* market has remained rather inflexible, thereby slowing down the innovation process in the field of navigation systems, and other in-vehicle (safety) applications. As pointed out in [2], based on current in-vehicle system market conditions and requirements of future applications such as advanced driver assistant systems (ADAS) and advanced navigation applications, it has been shown that in-vehicle applications will significantly benefit from up-to-date maps.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1097–1104, 2007.
© Springer-Verlag Berlin Heidelberg 2007

Thus, flexible, adaptive navigation using data that can be updated and personalized individually as well as extended for the public benefit still remains an interesting research topic with good chances for commercial value at a later stage.

2 Benefits from Our Approach

The only current solution to update map data of navigation systems is to supply update CDs or DVDs with completely or partially new map content (cf. a respective initiative of Navteq and DaimlerChrysler shown in 2006 at ITS in San Francisco for a partial update), which are available in low frequency (every 6–12 months). All these updates are created offline; vehicle position data while moving are solely used for determining the current vehicle status, but not for determining possible map adjustments, and information derivation.

Thus, in our approach we concentrate on enabling the integration of various types of up-to-date recorded map data delivered by the recording entities into one basic set and, thereby, the computation of the most probable digital map representing the real world road network including the related attributes. This results in the following major improvements:

- *Better Maps*: By individual additions of missing map segments as well as broadcasting them via a server (after verification), maps are being kept highly up-to-date, particularly for highly frequented new road segments.
- *Better Guidance*: Besides generalized optimization criteria (shortest route, scenic route, etc.), user-specific criteria can be used for route calculation allowing a highly automated personalization of the navigation system (e.g., routes considered safe by a driver).
- *Better Vehicle Efficiency*: Additional map information can be used to adapt the vehicle electronics to the environment and conditions. Curve radii, for example can be used to adjust the electronic stability programs, or slope information may help adjusting the fuel injection system for optimal engine performance.
- *Intelligent ADAS*: Currently, ADAS rely mostly on real-time input sources, like sensors or cameras. A vehicle navigation system itself (being an ADAS) can also be a source of valuable information for other ADAS on board, when sharing the acquired data in the form of an "electronic horizon" that contains up-to-date road geometry and related attributes ahead of the vehicle.

Our approach represents an important feature for the ubiquitous availability and quality of the map databases for future in-vehicle applications and it will even enable new functionality.

3 System Concepts Overview

3.1 Constraints

With respect to the fact that the system should preferably be widely used by the public at a later stage and the data will be broadcast via a server in order

to provide incremental updates for the public benefit, we have to consider the following quality criteria for a successful solution:

- *Simplicity*: The system will only obtain acceptance, if the hardware requirements and the required user interaction remain at minimum level.
- *System-independent data exchange interface*: The data exchange interface used for the distribution of the incremental updates must be system-independent and self-descriptive, due to the various data formats and structures used in proprietary navigation systems.
- *Turning quantity into quality*: In our approach, we assume that a satisfying data quality can be achieved by deducing the optimal average from a vast quantity of possibly noisy, low-cost-based data that denotes a specific amount of information.

The recording entities must be able to record track data and related attributes in the background, requiring only minimum user interaction, thereby enabling the user to reuse the data in-the-loop, i.e. for map data improvement solely for the entity that collects the data (where the person collecting and using the data generally is the same and, thus can rather easily be made responsible for using the data at her/his own risk), or to share the data via the central server.

3.2 Components

All common navigation systems are based on closed source. Therefore an in-the-loop integration of up-to-date map components can be implemented exclusively by the producer. Thus, our generic approach focuses on the server-based data integration and redistribution. However, we have implemented a prototypic proprietary solution for the in-the-loop use case as well, which is already integrated into the test bed of a current Siemens car navigation system, and has proven its ability to extend the digital map data by roads driven that are not contained in the available data (like the "S1" circumnavigation freeway south of Vienna, Austria; cf. Fig. 1).

The system's general architecture is based on two main system components: the *navigation entities*, and the *central server* (cf. Fig. 2). In addition, adequate *data exchange interfaces* are also part and parcel of the system. These components are presented in more detail in the following sections.

3.3 Navigation Entities

Basically, all kinds of GPS-enabled mobile devices can be used for recording GPS tracks to be uploaded to the server, if the output data format is among the supported ones. Most of the *waypoint-based navigation systems* (mostly for outdoor navigation, e.g., Garmin devices) support track recording ex works and can't be adapted in any way. Therefore, in order to prove our system's concept as a whole on concrete examples, we utilize two *map-based navigation systems* for recording GPS tracks including related attributes as well as for adapting the available map data based on incremental updates distributed by the central

server: an adapted Siemens car navigation system and a PDA-based mobile navigation and recording system developed from scratch.

Our PDA-system is based on the Microsoft .NET Compact Framework 2.0 and comprises the following modules: GPS interface, core, GUI, and data interface. The latter provides a self-implemented O/R-mapping abstraction layer above an exchangeable relational database (we utilize SQLite for this purpose), enabling an object-oriented parceling and adaptation of the map data. The core module includes the basic geometry, track recorder, navigation, update management, and event management submodules, and thereby integrates the entire application functionality.

3.4 Central Server

This component comprises the ability to receive the data from the recording entities, process them, and generate incremental map updates to be offered via a public service. The basic use case results in the following data flow:

- The *communication interface module* administrates multiple connection interfaces (WWW portal, wireless connections, etc.) for receiving the data. The parsers and converters for the most common track data formats (e.g., NMEA 0183, G7ToWin, GarFile), which can be embedded in the integrative XML-based upload files, are consolidated in a separate submodule implementing the Factory-Method design pattern.
- After extracting the raw track data to internal data structures pooled in the *geometry module*, data processing is done in the *core module*, comprising data processing, verification, and map update management.
- The data processing includes filtering out the outlier errors and insignificant data, converting the track data to a navigable graph, assigning the related attributes, merging of multiply recorded data, and enhancing an existing navigation graph. These steps necessitate a significant effort in finding suitable algorithmic solutions and are presented in more detail in Sect. 4.
- When not used in the loop, solid validation of the data gathered is necessary. Although automated plausibility checks can be performed on the data and data sets can be classified according to the overall source reliability, manual inspection will never be able to be eliminated completely. Therefore, we developed a *GUI module* that enables visual evaluation, editing, and further management of the digital map data on the central server.
- In order to administer a vast quantity of incremental map updates, we have developed a concept based on metadata-based versioning which, however, is not implemented yet.

All of the above modules rely on the central *database interface module* which, similar to the solution in our PDA-system, provides an object-oriented abstraction layer above an exchangeable relational database, allowing for manageable parceling and adaptation of the map data.

3.5 Data Exchange Interfaces

In order to enable the exchange of attributes and other metadata related to the GPS tracks, we defined an XML-based format with embedded GPS track data in one of the supported formats for uploading the data to the server. Thereby the users can either upload their data directly from their navigation system (if supported, like in our PDA-system, or upload the raw output data via a WWW portal and specify the metadata and attributes using an interactive web interface.

We base our incremental data update strategy on the specification of the EU-supported project "ActMAP", which specifies the strategies and the technical specifications for online incremental updating of in-vehicle digital map databases. The update exchange format specified by ActMAP is a standardized intermediate format based on XML designed for exchanging map updates between the proprietary formats of the map update suppliers and the map update users. Its logical data model is based on the ISO standard "Geographic Data Files" (GDF) and the XML schema. For further details on ActMAP, we refer to [9].

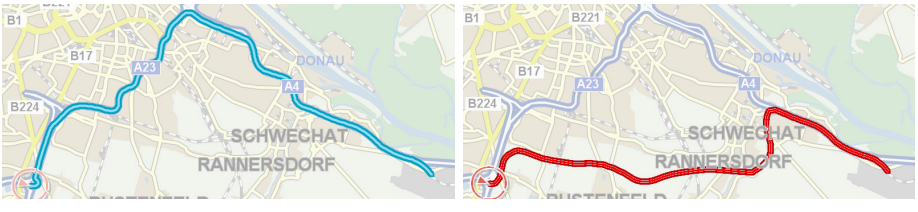


Fig. 1. In-the-loop reuse of recorded data in a Siemens car navigation system (left: route calculated using original map; right: route calculated using extended map)

4 Algorithmic Solution

When a GPS-enabled device is used to record a sequence of GPS points, this sequence can be used for an iterative point-to-point navigation allowing the retracing of a stored track from one GPS point to the next. Whereas such a *GPS track* is comprised of a sequence of GPS points, we define a *GPS route* (short: route) to be a linear graph of vertices and edges, where the vertices are defined by the GPS points, and the edges connect two subsequent GPS points and may contain additional information (like slope, terrain type, etc.).

Depending on the type of entity used for recording the route (vehicle, pedestrian, bicyclist, etc.), a *mobility model* can be defined describing maximum velocity, maximum curve radii, etc. According to this model, the most probable GPS route within the limits of the GPS track data can be determined.

If the same geographic path is recorded several times by the same entity or different ones, the most probable course of the real path or road can be determined by a suitable weighted averaging method. We use Kalman filtering [5] for

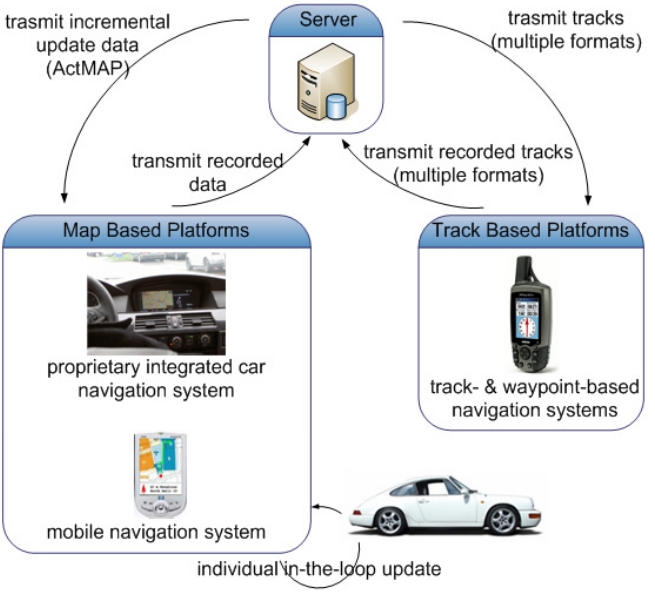


Fig. 2. System architecture overview

this purpose, defining a suitable *terrain model* (taking into account environment-specific factors that influence the GPS data quality) and a *receiver model* (modeling the specifics of the GPS device) in order to increase model accuracy (see [8] for further details).

For enhancing existing map graphs, we utilize a sub-graph matching ("registration") algorithm that was originally developed for analyzing blood vessels of the human brain [4] and has been adapted for geographic data by us. We use an adapted version of the clustering approach presented in [11] and modified in [1], combined with a self-implemented plane-sweep method based on the proposal in [3]. For human intervention we devised a graphic visualization tool in order to better understand the topology of the individual case.

5 ADAS Data

The benefit from adapting the geographical road network data can be multiplied by considering that valuable data from various sources (e.g., gyroscope, odometer, trip computer, Controller Area Network, etc.) are available to the integrated vehicle navigation systems in real time. By defining a set of logical rules for implying utilizable information based on the combination of the data with the vehicle position information, such data (we generalize them as ADAS data) can be a valuable source of information for the navigation system, and for other in-vehicle systems on board. In our GPS-track-related data classification model, we distinguish the following domains:

- *user-induced* (e.g. driver's preferred route),
- *condition-induced* (e.g. road condition),
- *induced by the means of travel* (e.g. motion model),
- *route-induced* (e.g. attributes like slope, placed objects like traffic signs).

In addition to the domain, we categorize potential ADAS data as *regional* (regarding whole road segments, e.g. road class), *discrete* (regarding a specific point within a road segment, e.g. traffic signs) and *continuous* (changing along a road segment, e.g. slope). Based upon the essential idea of our approach described in Sect. 2, we differentiate between an *objective* ("general", e.g. adding new road segments, traffic signs, etc.), and a *subjective* ("personal", e.g. adjusting the weighting of road segments, which belong to a driver's preferred route) adaptation of navigation data.

Since vehicle navigation systems in general would also benefit from additional attribute data related to the map graph already available, application scenarios include the addition of slope information to road segments, adjustable segment weighting against date/time constellation susceptible to traffic jams, etc.

However, due to the proprietary, mostly inflexible data structures, which are optimized for performance and not for extensibility, the integration of additional data might prove to be rather complex, if feasible at all. Let us illustrate this at the example of integrating slope data: Currently available navigation systems do not contain any altitude information. By utilizing the altitude included in each recorded GPS point, slope information can be easily derived for all driven road segments, but the best way for its integration must be studied individually for each proprietary data model.

When enabling the adaptation of the map data to the personal preference (based on infering the preference from the driver's behavior), many application scenarios can be imagined, e.g., an automatic recognition of "unpopular" areas or crossroads, preferred POIs, etc.

As already pointed out in Subsect. 3.2, since all common navigation and other in-vehicle systems are based on closed source, personalization and adaptation features, as proposed above, can only be implemented by their producers.

6 Conclusion

Fields of application of our approach include:

- *personalized adaptation* of navigation data,
- *optimization* of fleet management,
- *harmonized navigation data* from different sources (pedestrians, bicyclists, automobilists etc.).

Our prototypes show that our approach constitutes a significant advantage with respect to route data collection for personal use as well as map portal servers and in the future may also allow for continuously updated and personalized map data of in-vehicle systems. Hence, fine-tuning of ADAS systems is also thinkable

and may lead to a considerable adaptation of the car behavior and performance to driver style, environment topography and conditions, for instance.

Moreover, our adaptable PDA-based navigation system is already being used in the field. Due to its modular architecture, its GUI can be tailored to the target group's needs, thus enabling future navigation modules to utilize our core module (see [7] for further fields of application).

Acknowledgments. Parts of this work have been financed by the Austrian government (FFG), grant no. 811406 (FHplus programme). Other financially contributing partners to this project are Siemens Austria and Intersport Austria. Know-how is also given by Siemens VDO, Regensburg, Germany.

The authors would like to particularly thank Charly Artmann of Siemens PSE Austria, and Markus Schupfner of Siemens VDO Regensburg for all their initiatives and support of the project.

References

1. Brüntrup, R., Edelkamp, S., Jabbar, S., Scholz, B.: Incremental Map Generation with GPS Traces. In: Proc. 8^{th} International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria, pp. 413–418 (September 13–16, 2005)
2. eSafety Implementation Road Map Work Group: Final Report and Recommendations of the Implementation Road Map Working Group. eSafety Forum WG for Directorate-General Information Society of the European Commission (2005)
3. Freiseisen, W., Pau, P.: A Generic Plane-Sweep for Intersecting Line Segments. Res. Inst. for Symbolic Comuptation, J.K. University Linz, Austria. RISC-Linz Report Series No. 98-18 (1998)
4. Friedl, W.: The Registration Process in Transcranial Doppler Sonography (in German), Master's Thesis, Department of Medical Software Engineering, Upper Austria University of Applied Sciences, Hagenberg, Austria (2004)
5. Grewal, M.S., Andrews, A.P.: Kalman Filtering: Theory and Practice Using MATLAB, 2^{nd} edn. John Wiley & Sons, Chichester (2001)
6. Hofmann-Wellenhof, B., Lichtenegger, H., Collins, J.: GPS – Theory and Practice, 5^{th} edn. Springer, New York (2001)
7. Mayr, H.: Model-Based Navigation Using GPS: One Step Closer to Intelligent, Incremental Maps. In: Proc. International Mediterranean Modelling Multiconference, Barcelona, Spain, pp. 641–646 (October 4-6, 2006)
8. Mayr, H.: I-Navigate: Intelligent, Self-adapting Navigation Maps. In: Proc. 14^{th} IEEE International Conference and Workshop on the Engineering of Computer Based Systems, Tucson (AZ), USA, pp. 397–402 (March 26-29, 2007)
9. Otto, H.-U., Beuk, L., Aleksić, M., Meier, J., Löwenau, J., Flament, M., Guarise, A., Bracht, A., Capra, L., Bruns, K., Sabel, H.: ActMAP ISO Input. Technical Report D 3.3, ERTICO (2004)
10. Quddus, M., Noland, R., Ochieng, W.: Validation of Map Matching Algorithms Using High Precision Positioning with GPS. J. of Navigation 58, 257–272 (2005)
11. Schroedl, S., Wagstaff, K., Rogers, S., Langley, P., Wilson, Ch.: Mining GPS Traces for Map Refinement. Data Mining and Knowledge Discovery 9, 59–87 (2004)

Road Approximation in Euclidean and v-Disparity Space: A Comparative Study*

Angel D. Sappa[1], Rosa Herrero[1], Fadi Dornaika[2], David Gerónimo[1], and Antonio López[1]

[1] Computer Vision Center, Edifici O Campus UAB
08193 Bellaterra, Barcelona, Spain
{asappa, rherrero, dgeronimo, antonio}@cvc.uab.es
[2] Institut Géographique National
94165 Saint Mandé, France
fadi.dornaika@ign.fr

Abstract. This paper presents a comparative study between two road approximation techniques—planar surfaces—from stereo vision data. The first approach is carried out in the v-disparity space and is based on a voting scheme, the Hough transform. The second one consists in computing the best fitting plane for the whole 3D road data points, directly in the Euclidean space, by using least squares fitting. The comparative study is initially performed over a set of different synthetic surfaces (e.g., plane, quadratic surface, cubic surface) digitized by a virtual stereo head; then real data obtained with a commercial stereo head are used. The comparative study is intended to be used as a criterion for fining the best technique according to the road geometry. Additionally, it highlights common problems driven from a wrong assumption about the scene's prior knowledge.

1 Introduction

Recently, several techniques relaying on stereo vision systems have been proposed in the literature for driver assistance applications. These techniques have to deal with classical 3D processing and modelling problems together with real time constraints. The latter have motivated the use of driver scene's prior knowledge in order to simplify the problem.

A common problem in every on-board vision system (monocular/stereo) is the real time estimation of position and orientation, related to the current 3D road plane parameters—the ego-motion problem (e.g., [1], [2], [3]). Note that since the 3D plane parameters are expressed in the camera coordinate system, the camera position and orientation are equivalent to the plane parameters. Different algorithms have been proposed in the literature for road approximation from

* This work has been partially supported by the Spanish Ministry of Education and Science under project TRA2004-06702/AUT. The first author was supported by The Ramón y Cajal Program. The fourth author was supported by Spanish Ministry of Education and Science grant BES-2005-8864.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1105–1112, 2007.
© Springer-Verlag Berlin Heidelberg 2007

stereo vision data. These approaches can be broadly classified into two categories depending on the space where the processing is performed: *v-disparity* space and *Euclidean* space.

v-disparity space based approaches, originally proposed by [4], obtain a road approximation by using the scene prior knowledge and avoiding 3D data point computation. Disparity values correspond to the differences between the right and left images, while the corresponding *v*-disparity representation is computed by accumulating the points with the same disparity value that occur on a given image line. The *v*-disparity space allows to represent in a compact way the current scene geometry. An appealing feature of *v*-disparity representations lie in the fact that a plane in the Euclidean space becomes a straight line in the *v*-disparity space. Therefore, plane fitting becomes segment fitting. Actually, instead of segment fitting the use of Hough transform has been proposed in [4]; since then also adopted by those techniques working in the *v*-disparity space. The Hough transform is used with a voting scheme to extract the largest sub-set of points that define a straight line; unlike fitting techniques that find the best straight line for the whole set of points (e.g., least squares fitting).

After the original proposal [4], several *v*-disparity based approaches have been developed for driver assistance: obstacle or pedestrian detection (e.g., [5], [6], [7]), atmospheric visibility measurement system [8], etc. Recently, the *v*-disparity approach has been extended to a *u*-*v*-disparity concept in [9]. In this new proposal, dense disparity maps are used instead of only relying on edge based disparity maps. Working in the disparity space is an interesting idea that is gaining popularity in on-board stereo vision applications, since planes in the original Euclidean space become straight lines in the disparity space. However, it should be noticed that this approach has been proposed under the assumption that the road geometry fit to a plane. If this assumption does not hold, the straight line extracted in the *v*-disparity space wouldn't correspond to the best fitted plane in the Euclidean space.

In turn, Euclidean space based approaches are focused on the use of 3D data points directly in the 3D space—dense or sparse representations. For instance, [10] proposes a road approximation technique that works in the Euclidean space, by using a method similar to the Hough transform over a lateral projection of the original 3D points. On the contrary, a least squares fitting approach is used in [11], by previously removing outliers. Hence, a real time estimation of on-board camera extrinsic parameters, related to the plane that minimizes the sum of the squares of the offsets ("the residuals") of the whole set of points, is obtained. In this case, road plane fitting is performed in the 3D Euclidean space by using a compact and structured representation of the raw input data points.

The current paper aims at comparing a *v*-disparity space based approach [4] with an Euclidean space based approach [11], without going into details on their corresponding implementations; the main objective is to study their validity, while at the same time conclusions are obtained.

The remainder of this paper is organized as follows. Section 2 briefly describes the *v*-disparity space based road approximation technique [4]. Then, section 3

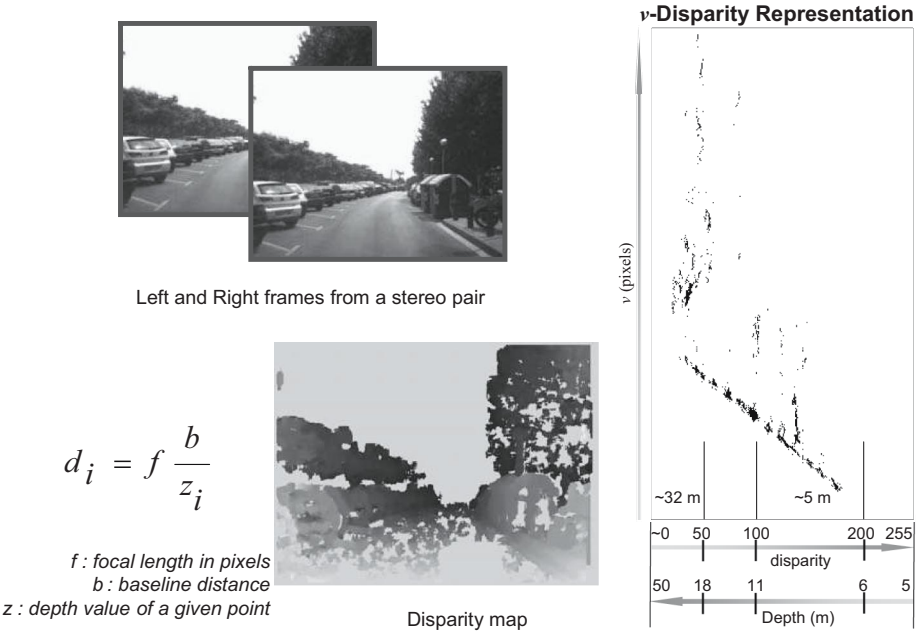


Fig. 1. ($left$) Disparity map from the left and right images of a given stereo head. ($right$) The corresponding v-disparity representation.

introduces the road approximation technique in Euclidean space [11]. Section 4 presents comparisons obtained with synthetic and real scenes. Finally, discussions and conclusions from the comparative study are given in section 5.

2 Road Approximation in v-Disparity Space

As mentioned above several approaches have been proposed based on the fact that a plane in the Euclidean space becomes a straight line in the v-disparity space. Up to our knowledge, all these approaches are based on the use of Hough transform for detecting a straight line, which is assumed to be the road plane projection in the v-disparity representation. Figure 1($right$) shows the v-disparity representation corresponding to the disparity map presented in Figure 1($left$).

Hough Transform (HT) is generally used in image processing for finding shapes in images or set of data points. The underlying principle of HT is that there is an infinite number of potential lines that pass through any point, each at a different orientation. Hence if all possible straight lines, for every data point in the image, are considered at once and represented in a (r, θ) space, which correspond to the angle and distance to the origin of a normal to the line in question. The position with most crosses at the (r, θ) space will define the straight line with most points. This voting scheme has been largely used for feature detection as well as for solving regression problems. For instance, HT has been used by [4],

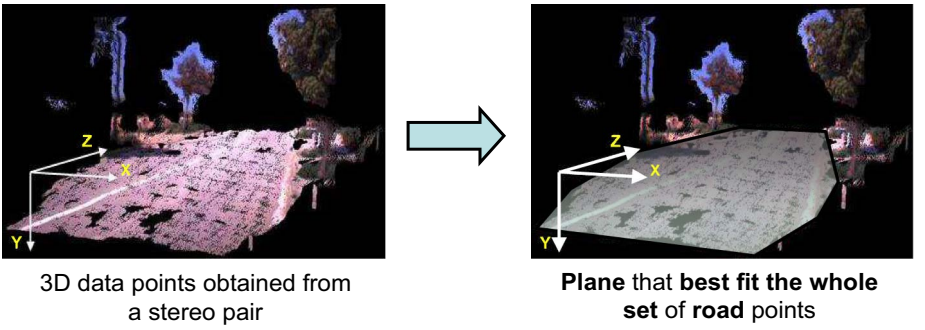


Fig. 2. Road approximation in Euclidean space

[5], [6] and [7] to find the the largest sub-set of points that define a straight line in the v-disparity space. This straight line corresponds to the 3D road plane in the Euclidean space.

3 Road Approximation in Euclidean Space

Approaches proposed to work in Euclidean space essentially consist of two stages. Initially, 3D data points are structured in such a way that a real time processing could be performed. Then, a fitting technique is used for finding the best surface that approximate those data points [12]. The least squares fitting technique is the simplest and most commonly applied procedure for finding the best-fitting surface to a given set of points. It minimizes the sum of the squares of the offsets of the points from the surface. The sum of squares of the offsets is used instead of the offset absolute values because this allows the residuals to be treated as a continuous differentiable quantity.

In [11], the first stage consists in generating a compact 2D representation of the original 3D data points—lateral view of 3D data points. Then a RANSAC based least squares approach is used for fitting a plane to the road candidate points selected in the 2D projection (YZ plane). Figure 2 presents a sketch of road approximation in Euclidean space.

4 Comparisons

The two approaches introduced above have been compared with different road geometries. Initially, synthetic data points were obtained by using a virtual stereo head. Then, data points from real scenes were considered by using a commercial stereo head.

4.1 Synthetic Data

A virtual stereo head has been defined by using two virtual pin-hole cameras. A similar configuration to the one provided by the commercial stereo head used

Table 1. Comparisons: **Synthetic Scenes**

SURFACE	v**-DISPARITY SPACE** [4] {Mean Sq./Max.} Error [m]	**EUCLIDEAN SPACE** [11] {Mean Sq./Max.} Error [m]
Plane	0 / 0	0 / 0
(with noise)	0.007 / 0.008	0.007 / 0.008
Quadratic (35 m at 1000m)	2.68 / 34.49	0.44 / 8.47
(with noise)	2.68 / 36.69	0.44 / 8.93
Quadratic (100 m at 1000m)	24.14 /103.48	3.98 / 25.41
(with noise)	24.12 / 110.29	3.98 / 26.99
Cubic (35 m at 1000m)	2.89 / 36.98	0.80 / 11.79
(with noise)	2.89 / 39.44	0.80 / 12.52
Cubic (100 m at 1000m)	20.61 / 98.63	5.69 / 31.45
(with noise)	20.60 / 105.38	5.69 / 33.49

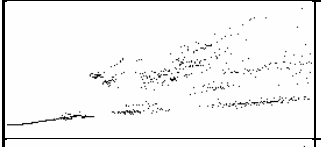
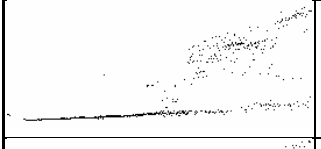
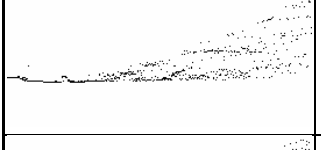
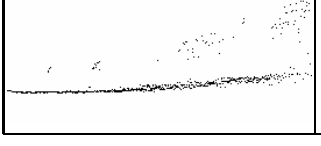
in the real data experiments has been adopted (see 4.2). Different surfaces have been considered and their corresponding images (left/right) have been used for computing disparity values. Disparity values were used for obtaining the corresponding v-disparity representation. For every pixel of the synthetic images 3D information was directly computed by finding the intersection between the ray that passes through that pixel (image plane) and the considered surface. Every surface was used twice; firstly, with the synthetic data computed as indicated above and secondly, by adding noise to the surface. The noise grows according to the depth trying to model accuracy of stereo vision systems—inversely proportional to depth (e.g., [13], [14]).

Following [4], the HT was used for obtaining road plane parameters from the straight line extracted in the v-disparity space. On the other hand the corresponding 3D data points were directly used for computing road plane parameters in the Euclidean space [11]. In both cases, the mean square error and maximum error between the computed planes and the set of 3D data points were computed and used as comparison criteria between the two techniques. Table 1 presents results obtained with different surfaces. As it was expected, both approaches give the same results when planar surfaces are considered (with/without added noise). On the contrary, smaller mean square errors and maximum errors are obtained by using a fitting scheme in the Euclidean space [11] instead of using HT in v-disparity space [4], when non-planar surfaces are considered.

4.2 Real Data

A commercial stereo vision system (`[www.ptgrey.com]` Bumblebee from Point Grey) was used. It consists of two Sony ICX084 Bayer pattern CCDs with 6mm focal length lenses. Bumblebee is a pre-calibrated system that does not require in-field calibration. The baseline of the stereo head is 12 cm and it is connected

Table 2. Comparisons: **Real Scenes**

Road Profile in Euclidean Space	Plane Fitting in v-**DISPARITY SPACE** [4] {Mean Sq./Max.} Error [m]	Plane Fitting in **EUCLIDEAN SPACE** [11] {Mean Sq./Max.} Error [m]
	0.94 / 8.51	0.66 / 5.35
	3.34 / 13.88	1.74 / 8.22
	2.58 / 16.58	0.96 / 11.27
	0.87 / 11.22	0.60 / 9.08

to the computer by a IEEE-1394 connector. Right and left color images were captured at a resolution of 640×480 pixels. Camera control parameters were set to automatic mode to compensate global changes in the light intensity. For every couple of right/left images the corresponding disparity image and 3D data were computed by using the provided 3D reconstruction software. Disparity values were used for computing the v-disparity representation. It is later on used by [4] to obtain the road plane parameters. On the other hand, 3D data points were used for fitting the road plane as indicated in [11]. As in the synthetic case, mean square errors and maximum errors were computed for comparing the obtained results. Table 2 presents some of the results obtained with different road geometries. It can be appreciated that in all the cases working in the Euclidean space, by using a fitting technique, gives better results than those obtained in v-disparity space, by using HT.

5 Discussions and Conclusions

This paper presents a comparative study between two different road approximation techniques generally used on stereo based driver assistance applications.

Although both approaches are valid for road plane extraction, working in the v-Disparity space could drive to wrong results do to the use of Hough transform with non-planar road geometries.

Although voting and fitting schemes would give the same result when a perfectly planar road is considered (ideal case) differences will appear when non planar roads are processed. It is easy to see that the segment that passes through more points will be obtained by using the HT, which does not necessarily correspond to the most representative one for the whole set of points.

The non-linear representation of disparity values (Figure 1($left$)) rises up as an additional drawback when v-disparity space is considered. Figure 1($right$) presents the v-disparity representation of a scene ranging between 5 and 50 meters in depth. Notice that less than a quarter of disparity values (from 0 up to 50, having a total span higher than 200 values) are used for representing more than 70% of depth values (distances from 18 up to 50 meters). This non-linear mapping makes that more attention is paid to nearest points, instead of considering all the points equally (almost half of disparity values, from 100 to 200 are used for representing distances in between 6 and 11 meters, about 11% of depth values). Recently, [15] has also noticed this drawback and proposed a measurement to estimate the quality of the v-disparity image according to road flatness. This image quality value is used for computing the on-board camera pitch orientation.

Although out of the scope of this comparative study, it should be mentioned that the extraction of planar representations with Hough transform in the v-Disparity space is faster than fitting in Euclidean space, since most of the CPU time required by 3D reconstruction algorithms is avoided.

We can conclude that [4] can be used in highways environments where road vertical curvature could be neglected. However, if the on-board stereo system is intended to be used on urban driver assistance applications [4] could drive to wrong results, particularly those urban scenarios in non-flat regions. In this case, a technique such as the one presented in [11] is better suited.

References

1. Bertozzi, M., Broggi, A.: GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. IEEE Trans. on Image Processing 7(1), 62–81 (1998)
2. Milella, A., Siegwart, R.: Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In: Proc. IEEE Int. Conf. on Computer Vision Systems, IEEE Computer Society Press, New York, USA (2006)
3. van der Mark, W., Fontijne, D., Dorst, L., Groen, F.: Vehicle ego-motion estimation with geometric algebra. In: Proc. IEEE Intelligent Vehicles Symposium, Versailles, France, pp. 58–63. IEEE Computer Society Press, Los Alamitos (2002)
4. Labayrade, R., Aubert, D., Tarel, J.: Real time obstacle detection in stereovision on non flat road geometry through 'V-disparity' representation. In: Proc. IEEE Intelligent Vehicles Symposium, Versailles, France, pp. 646–651. IEEE Computer Society Press, Los Alamitos (2002)

5. Bertozzi, M., Binelli, E., Broggi, A., Del Rose, M.: Stereo vision-based approaches for pedestrian detection. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, IEEE Computer Society Press, San Diego, USA (2005)
6. Broggi, A., Fascioli, A., Fedriga, I., Tibaldi, A., Del Rose, M.: Stereo-based preprocessing for human shape localization in unstructured environments. In: Proc. IEEE Intelligent Vehicles Symposium, pp. 410–415. IEEE Computer Society Press, Columbus, OH, USA (2003)
7. Labayrade, R., Aubert, D.: A single framework for vehicle roll, pitch, yaw estimation and obstacles detection by stereovision. In: Proc. IEEE Intelligent Vehicles Symposium, pp. 31–36. IEEE Computer Society Press, Columbus, OH, USA (2003)
8. Hautière, N., Labayrade, R., Aubert, D.: Real-time disparity contrast combination for onboard estimation of the visibility distance. IEEE Trans. on Intelligent Transportation Systems 7(2), 201–212 (2006)
9. Hu, Z., Uchimura, K.: U-V-Disparity: An efficient algorithm for stereovision based scene analysis. In: Proc. IEEE Intelligent Vehicles Symposium, pp. 48–54. IEEE Computer Society Press, Las Vegas, USA (2005)
10. Nedevschi, S., Danescu, R., Frentiu, D., Graf, T., Schmidt, R.: High accuracy stereovision approach for obstacle detection on non-planar roads. In: Proc. IEEE Inteligent Engineering Systems, Cluj Napoca, Romania, pp. 211–216. IEEE Computer Society Press, Los Alamitos (2004)
11. Sappa, A., Gerónimo, D., Dornaika, F., López, A.: On-board camera extrinsic parameter estimation. Electronics Letters 42(13), 745–747 (2006)
12. Rousseeuw, P., Leroy, A.: Robust Regression and Outlier Detection. John Wiley & Sons, New York (1987)
13. Toulminet, G., Bertozzi, M., Mousset, S., Bensrhair, A., Broggi, A.: Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis. IEEE Trans. on Image Processing 15(8), 2364–2375 (2005)
14. Scherba, D., Bajcsy, P.: Depth map calibration by stereo and wireless sensor network fusion. In: Proc. IEEE Int. Conf. on Information Fusion, Philadelphia, USA, pp. 1540–1547. IEEE Computer Society Press, Los Alamitos (2005)
15. Broggi, A., Caraffi, C., Porta, P., Zani, P.: The single frame stereo vision system for reliable obstacle detection used during the 2005 DARPA Grand Challenge on TerraMax. In: Proc. IEEE Int. Conf. on Intelligent Transportation Systems, Toronto, Canada, pp. 745–752. IEEE Computer Society Press, Los Alamitos (2006)

Vision-Based Blind Spot Detection Using Optical Flow

M.A. Sotelo[1], J. Barriga[1], D. Fernández[1], I. Parra[1],
J.E. Naranjo[2], M. Marrón[1], S. Alvarez[1], and M. Gavilán[1]

[1] Department of Electronics, University of Alcalá, Alcalá de Henares, Madrid, Spain
{sotelo, llorca, marta, parra}@depeca.uah.es
[2] Industrial Automation Institute, CSIC, Arganda del Rey, Madrid, Spain
jnaranjo@iai.csic.es

Abstract. This paper describes a vision-based system for blind spot detection in intelligent vehicle applications. A camera is mounted in the lateral mirror of a car with the intention of visually detecting cars that can not be perceived by the vehicle driver since they are located in the so-called blind spot. The detection of cars in the blind spot is carried out using computer vision techniques, based on optical flow and data clustering, as described in the following lines.

Keywords: Computer Vision, Optical Flow, Blind Spot, Intelligent Vehicles.

1 Introduction

A vision-based blind spot detection system has been developed for Intelligent Vehicle Applications. The main sensor for providing detection of cars is vision. Images are analyzed using optical flow techniques [1] in order to detect pixels that move in the same direction as the ego-vehicle. Pixels producing movement as described are grouped following the clustering techniques described in [2]. The resulting clusters are considered as potential vehicles overtaking the ego-vehicle. A double-stage detection mechanism has been devised for providing robust vehicle detection. In a first stage, a pre-detector system computes the mass center of the resulting clusters and determines whether the detected cluster is a potential vehicle according to the size of detected pixels.

In a second stage, another detector looks for the appearance of vehicles frontal parts. Any object looking like the frontal part of a vehicle is considered as a potential vehicle, whenever the mass center pre-detector triggers the pre-detection signal. Thus, a sufficiently big object in the image plane, producing optical flow in the same direction as the ego-vehicle, and exhibiting a part similar to the frontal part of a car is validated as a car entering the blind spot. The position of the vehicle in the image plane is computed and tracked using a Kalman Filter. Tracking continues until the vehicle disappears from the scene, and an alarm signal is triggered indicating the driver that a vehicle has entered the blind spot zone.

2 System Description

The description of the algorithm is provided in figure 1 in the form of flow diagram. As can be observed, there are several computation steps based on optical flow com-

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1113 – 1118, 2007.
© Springer-Verlag Berlin Heidelberg 2007

puting at image level, pixel-wise clustering, analysis of clusters and final vehicle detection. As previously stated, the system relies on the computation of optical flow using vision as main sensor providing information about the road scene. In order to reduce computational time, optical flow is computed only on relevant points in the image. These points are characterized for exhibiting certain features that permit to discriminate them from the rest of point in their environment. Normally, these salient features have prominent values of energy, entropy, or similar statistics. In this work, a salient feature point has been considered as that exhibiting a relevant differential value. Accordingly, a Canny edge extractor is applied to the original incoming image. Pixels providing a positive value after the Canny filter are considered for calculation of optical flow. The reason for this relies on the fact that relevant points are needed for optical flow computation since matching of the points have to be done between two consecutive frames.

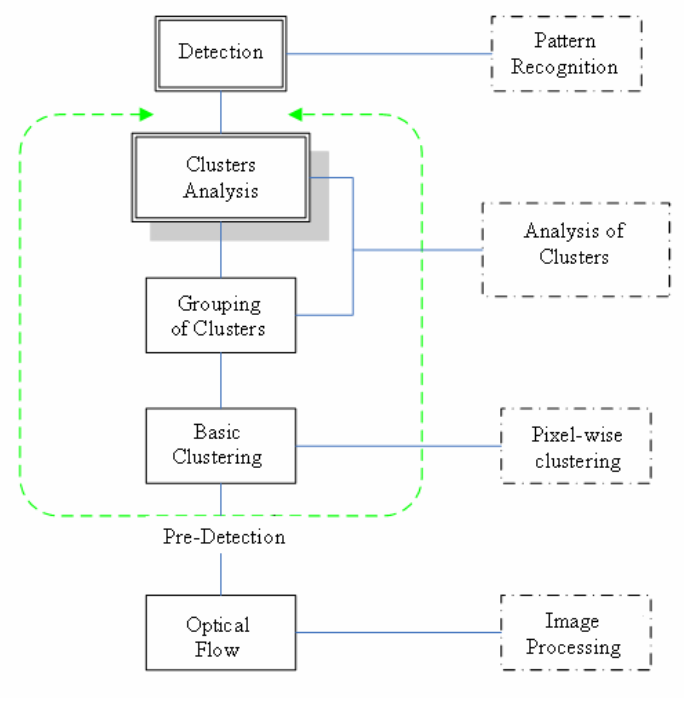


Fig. 1. Flow Diagram of the Blind-spot detection algorithm

After that, Canny edge pixels are matched and grouped together in order to detect clusters of pixels that can be considered as candidate vehicles in the image. Classical clustering techniques are used to determine groups of pixels, as well as their likelihood to form a single object. Figure 2 depicts a typical example of matched points after computing optical flow and performing pixels clustering.

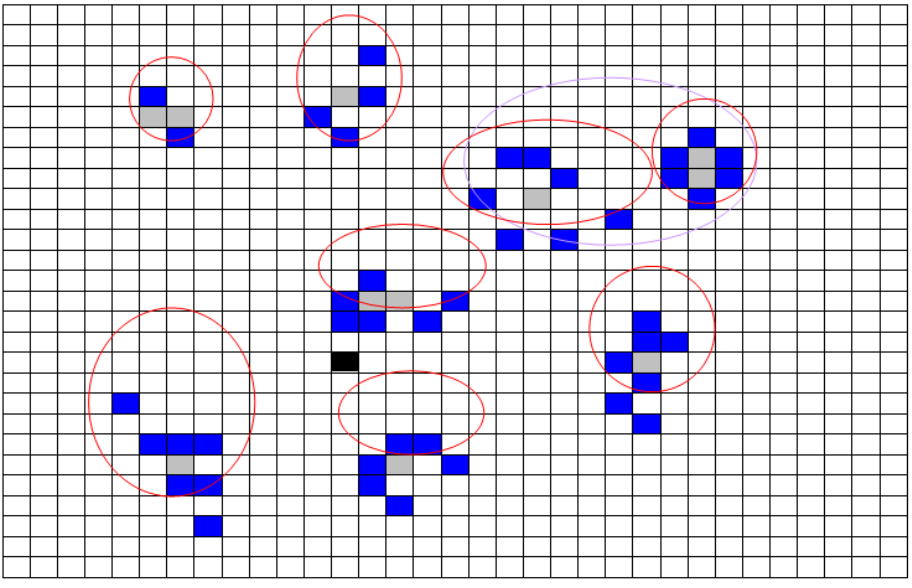


Fig. 2. Clustering of pixels providing relevant optical flow

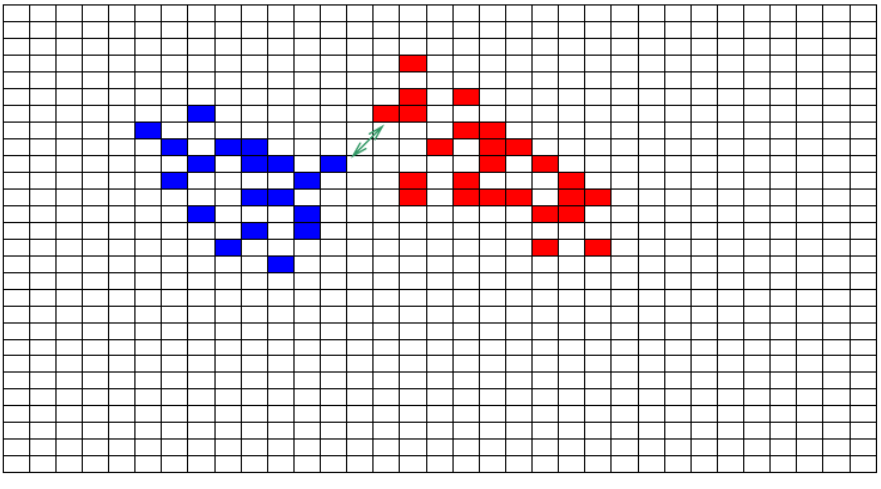


Fig. 3. Grouping of close clusters in a second clustering stage

Pixels in blue represent edge points that have produced relevant optical flow in two consecutive frames. Red ellipses stand for possible groups (clusters) of objects. Violet ellipses represent ambiguous groups of objects that could be possibly split in two. Gray pixels represent the mass center of detected clusters. Even after pixels clustering, some clusters can still be clearly regarded as belonging to the same real object. A second grouping stage is then carried out among different clusters in order to determine which of them can be further merged together into a single blob. For this purpose, simple distance criteria are considered. As depicted in figure 3, two objects that

are very close to each other are finally grouped together in the same cluster. The reason for computing a two-stage clustering process relies on the fact that by selecting a small distance parameter in the first stage interesting information about clusters in the scene can be achieved. Otherwise, i.e. using a large distance parameter in single clustering process, very gross clusters would have been achieved, losing all information about the granular content of the points providing optical flow in the image.

The selected clusters constitute the starting point for locating candidate vehicles in the image. For that purpose, the detected positions of clusters are used as a seed point for finding collection of horizontal edges that could potentially represent the lower part of a car. The candidate is located on detected horizontal edges that meet certain conditions of entropy and vertical symmetry. Some of the most critical aspects in blind spot detection are listed below:

1. Shadows on the asphalt due to lampposts, other artifacts or a large vehicle overtaking the ego-vehicle on the right lane.
2. Self-shadow reflected on the asphalt (especially problematic in sharp turns like in round-about points), or self-shadow reflected on road protection fences.

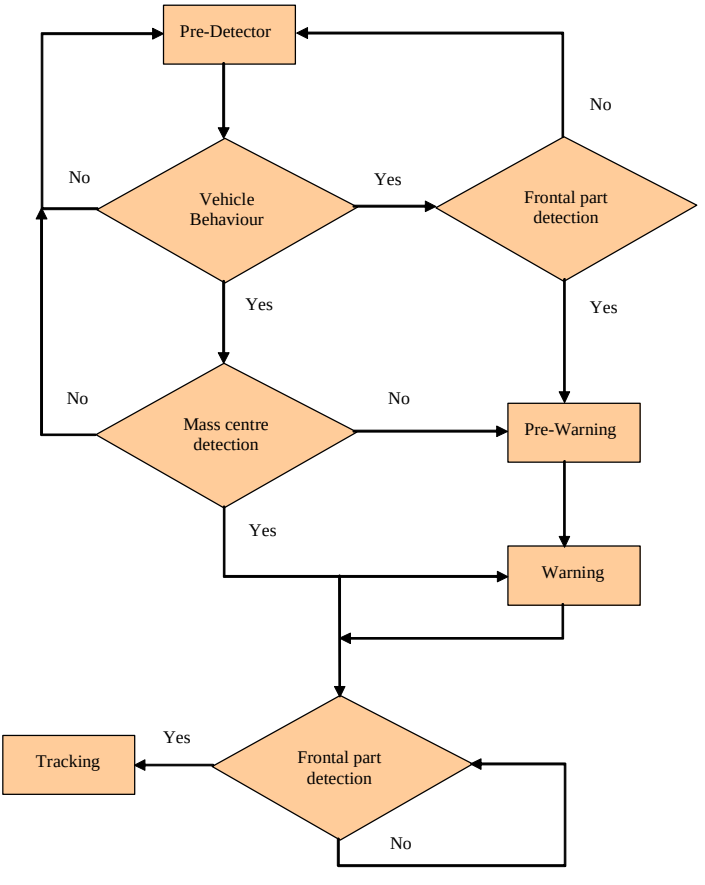


Fig. 4. Pre-detection Flow Diagram

3. Robust performance in tunnels.
4. Avoiding false alarms due to vehicles on the third lane.

The flow diagram of the double-stage detection algorithm is depicted in figure 4. As can be observed, there is a pre-detector that discriminates whether the detected object is behaving like a vehicle or not. If so, the frontal part of the vehicle is located in the Region Of Interest and a pre-warning is issued. In addition, the vehicle mass centre is computed. In case the frontal part of the vehicle is properly detected and its mass centre can also be computed a final warning message is issued. Vehicle tracking starts at that point. Tracking is stopped when the vehicle gets out of the image. Some times, the shadow of the vehicle remains in the image for a while after the vehicle disappears from the scene, provoking the warning alarm to hold on for 1 or 2 seconds. This is not a problem, since the overtaking car is running in parallel with the ego-vehicle during that time although it is our of the image scene. Thus, maintaining the alarm in such cases turns out to be a desirable side-effect.

After locating vehicle candidates, these are classified using a SVM classifier previously trained with samples obtained from real road images.

3 Implementation and Results

A digital camera was mounted in the lateral mirror of a real car equipped with a Pentium IV 2.8 GHz PC running Linux Knoppix 3.7 and OpenCV libraries 0.9.6. The car



Fig. 5. Example of blind spot detection in a sequence of images. The indicator in the upper-right part of the figure toggles from green to blue when a car is detected in the blind spot.

was manually driven for several hours in real highways and roads. After the experiments, the system achieved a detection rate of 99% (1 missing vehicle), producing 5 false positive detections. Figure 5 shows an example of blind spot detection in a sequence of images. The indicator depicted in the upper-right part of the figure toggles from green to blue when a vehicle enters the blind spot area (indicated by a green polygon). A blue bounding box depicts the position of the detected vehicle.

Our current research focuses on the development of SVM-based vehicle recognition for increasing the detection rate and decreasing the false alarm rate, as demonstrated in [3], where SVM was used for vehicle detection in an ACC application.

Acknowledgments. This work has been funded by Research Project CICYT DPI2005-07980-C03-02 (*Ministerio de Educación y Ciencia, Spain*).

References

1. Giachetti, A., Campani, M., Torre, V.: The use of optical flow for road navigation. IEEE Transactions on Robotics and Automation 14(1) (1998)
2. Kailunailen, J.: Clustering algorithms: basics and visualization. Helsinki University of Technology. Laboratory of Computer and Information Science. T61.195, Special Assignment 1 (2002)
3. Sotelo, M.A., Nuevo, J., Ocaña, M., Bergasa, L.M.: A Monocular Solution to Vision-Based ACC in Road Vehicles. In: Moreno Díaz, R., Pichler, F., Quesada Arencibia, A. (eds.) EUROCAST 2005. LNCS, vol. 3643, pp. 507–512. Springer, Heidelberg (2005)

Ego-Motion Computing for Vehicle Velocity Estimation

M.A. Sotelo[1], R. Flores[1], R. García[1], M. Ocaña[1], M.A. García[1], I. Parra[1], D. Fernández[1], M. Gavilán[1], and J.E. Naranjo[2]

[1] Department of Electronics, University of Alcalá, Alcalá de Henares, Madrid, Spain
{sotelo, rflores, mocana, garrido, parra, llorca}@depeca.uah.es
[2] Industrial Automation Institute, CSIC, Arganda del Rey, Madrid, Spain
jnaranjo@iai.csic.es

Abstract. In this paper, we present a method for computing velocity using a single camera onboard a road vehicle, i.e. an automobile. The use of computer vision provides a reliable method to measure vehicle velocity based on ego-motion computation. By doing so, cumulative errors inherent to odometry-based systems can be reduced to some extent. Road lane markings are the basic features used by the algorithm. They are detected in the image plane and grouped in couples in order to provide geometrically constrained vectors that make viable the computation of vehicle motion in a sequence of images. The applications of this method can be mainly found in the domains of Robotics and Intelligent Vehicles.

Keywords: Vision, Ego-motion, Velocity Estimation, Intelligent Vehicles.

1 Introduction

Accurate estimation of the vehicle ego-motion with regard to the road is a key element for computer vision-based assisted driving systems. In this method, we propose the use of a single camera onboard a road vehicle in order to provide an estimation of its longitudinal velocity by computational means. The main advantage derived from the use of computer vision for ego-motion computation is the fact that vision is not subject to slippery, contrary to odometry-based systems. We propose to obtain couples of road features, mainly composed of road markings, as the main source of information for computing vehicle ego-motion. Additionally, the use of lane markings allows avoiding the use of complex direct methods [1], [2], [3] for motion estimation. Instead, motion stereo techniques are considered. This technique has previously been deployed in the field of indoor robotics [4]. The method is based on sampling a dynamic scene rapidly (e.g., 25 images per second) and measuring the displacements of features relative to each other in the image sequence.

2 System Description

In outdoor scenes where many artificial objects and structures exist, a couple of static points that belong to the same object and are equally distant from the image plane

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1119 – 1125, 2007.
© Springer-Verlag Berlin Heidelberg 2007

may be observed and measured simultaneously. In particular, the left and right edges of lane markings constitute a clear example of coupled points that can be used for computing vehicle ego-motion using perspective projection laws. Let us, then, assume that there are two road points, P_1 and P_2, with coordinates (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2), where Z stands for the point depth (distance from the image plane). Let us assume that $Z_1=Z_2=Z$, which means that both points are equally distant from the image plane. The coordinates of the points in the image plane, p_1 and p_2, can then be computed as

$$\begin{aligned} p_1 &= \left(u_c + f_u \cdot \frac{X_1}{Z}, v_c + f_v \cdot \frac{Y_1}{Z} \right) \\ p_2 &= \left(u_c + f_u \cdot \frac{X_2}{Z}, v_c + f_v \cdot \frac{Y_2}{Z} \right) \end{aligned} \tag{1}$$

where u_c and v_c represent the coordinates of the principal point in the image plane (optical center), and f_u and f_v are the camera focal length, given in pixels units, in the u (horizontal) and v (vertical) axes, respectively. Let $B=|X_1-X_2|$ be the horizontal distance between the road points and $b=|x_1-x_2|$ the horizontal distance between the corresponding image points. Based on the previous statement, $b=f_u \cdot B/Z$. Finally, let us consider that the camera is translated causing the two road points to move relative to the camera with the velocity $(dX/dt, dY/dt, dZ/dt)$ while f_u and B remain constant. In general, the derivative of b with respect to time can be computed as

$$\frac{db}{dt} = \frac{db}{dZ} \cdot \frac{dZ}{dt} = -\frac{f_u B}{Z^2} \cdot \frac{dZ}{dt} = -\frac{b}{Z} \cdot \frac{dZ}{dt} \tag{2}$$

For a couple of road points, the distance from the image plane Z can be computed under the planar road assumption as follows

$$\begin{aligned} \theta &= \tan^{-1}\left(\frac{H}{Z} \right) \\ v &= f_v \cdot \tan(\theta - \alpha) \end{aligned} \tag{3}$$

where α stands for the camera pitch angle with respect to the horizontal line parallel to the road, v is the vertical coordinate of the coupled road points in the image plane, and H is the camera height with respect to the road plane. Let us remark that coordinate v can be directly measured from the image, while parameters H and α are supposed to be known.

Based on relations (2) and (3), an equation can be formulated for each couple i of road points equally distant from the image plane. Equation (4) shows a mathematical relation from which vehicle velocity $(v=dZ/dt)$ can be computed.

$$v = \frac{dZ}{dt} = -\frac{Z_i}{b_i} \cdot \frac{db_i}{dt} \tag{4}$$

Let N_t represent the number of road point couples found by the algorithm at frame *t*. The optimal estimation of vehicle velocity *v* can be done by optimizing the system formed by the N_t equations that can be written at each iteration of the algorithm. Based on the previous statement, the problem can be mathematically formulated as the minimization of the estimation square error *SE*, represented in equation 5.

$$SE = \frac{1}{N_t} \cdot \sum_{i=1}^{N_t} \left(\widehat{b_i} - b_{i,t} \right)^2 \tag{5}$$

where $\widehat{b_i}$ represents the estimation of *b* for couple *i*, and $b_{i,t}$ stands for the measurement of *b* for couple *i* at frame *t*. This criteria leads to the final value provided in equation (6).

$$v_t \approx \frac{\sum_{i=1}^{N_t} \left(\frac{db_{i,t}}{dt} \right) \cdot \frac{b_{i,t-1}}{Z_{i,t-1}}}{\sum_{i=1}^{N_t} \left(\frac{b_{i,t-1}}{Z_{i,t-1}} \right)^2} \tag{6}$$

where $b_{i,t-1}$ represents the measurement of *b* for couple *i* at frame *t-1*, and $Z_{i,t-1}$ stands for the depth measurement for couple *i* at frame *t-1*.

3 Extension of the Method - 3D Visual Odometry

We propose an extension of the method for ego-motion computing based on stereo-vision, achieving what is known as 3D visual odometry. The use of stereo-vision has the advantage of disambiguating the 3D position of detected features in the scene at a given frame. Based on that, feature points are matched between pairs of frames and linked into 3D trajectories. The idea of estimating displacements from two 3-D frames using stereo vision has been previously used in [5] [6] and [7]. The resolution of the equations of the system at each frame is carried out under the non-linear, photogrametric approach using RANSAC. This iterative technique enables the formulation of a robust method that can ignore large numbers of outliers as encountered in real traffic scenes. The resulting method is defined as visual odometry and can be used in conjunction with other sensors, such as GPS, to produce accurate estimates of the vehicle global position. The mathematical details of the method are provided in [8]. The obvious application of the method is to provide on-board driver assistance in navigation tasks, or to provide a means for autonomously navigating a vehicle. The method has been tested in real traffic conditions without using prior knowledge about the scene or the vehicle motion. We provide examples of estimated vehicle trajectories using the proposed method and discuss the key issues for further improvement.

In each frame, Harris corners are detected, since this type of point feature has been found to yield detections that are relatively stable under small to moderate image distortions. As stated in [6], distortions between consecutive frames can be regarded as fairly small when using video input. The feature points are matched at each frame, using the left and rights image of the stereo-vision arrangement, and between pairs of frames. Features are detected in all frames and matches are allowed only between features. A feature in one image is matched to every feature within a fixed distance from it in the next frame, called disparity limit. For the sake of real-time performance, matching is computed over a 7x7 window. Among the wide spectrum of matching techniques that can be used to solve the correspondence problem we implemented the *Zero Mean Normalized Cross Correlation* (ZMNCC) because of its robustness.

As the window size decreases, the discriminatory power of the area-based criterion gets decreased and some local maxima appear in the searching regions. On the contrary, an increase in the window size causes the performance to degrade due to occlusion regions and smoothing of disparity values across boundaries. In consequence, the correspondences yield some outliers. According to the previous statements, some filtering criteria are needed in order to provide outliers rejection. In order to minimize the number of outliers, mutual consistency check is used. Accordingly, only pairs of features that yield mutual matching are accepted as a valid match. It is important to remark that mutual consistency check can be accomplished without computing correlations more than once. The accepted matches are used both in 3D feature detection (based on stereo images) and in feature tracking (between consecutive frames). Figure 1 depicts an example of features detection and tracking using Harris detector, ZMNCC matching technique, and mutual consistency check.

4 Implementation and Results

The algorithm was implemented on a PC onboard a real automobile in a test circuit. A Firewire camera was mounted on the test car, providing 640x480 Black&White images in IEEE 1394 format. The couples of road points detected by the algorithm in a real experiment are depicted in green on the left hand side of Figure 1. It must be remarked that the correspondence of road points between two consecutive images is carried out by implementing an optical flow. In the same figure, the instantaneous estimation of vehicle velocity at the current frame is provided (37.24 km/h), as well as the accumulated length of the path run by the car (292.78m in this example). Similarly, the estimation of vehicle velocity is provided in the right hand side of Figure 2 for the complete duration of the experiment. The vertical axis represents vehicle velocity in km/h. The red curve depicts vehicle velocity estimation without filtering, while the blue curve depicts vehicle velocity estimation using a kalman filter. The final result issued by the algorithm demonstrated to be very similar to the vehicle velocity measured by odometry means (around 40 km/h).

At present, the estimation of vehicle velocity is being used in the prediction stage of kalman filtering in Lane Departure Warning (LDW) Systems developed by the authors. Similarly, the estimation of vehicle ego-motion is currently being extended to a 6-component vector providing the complete ego-motion information, including vehicle longitudinal and angular displacement in X, Y, and Z. Figure 3 depicts an

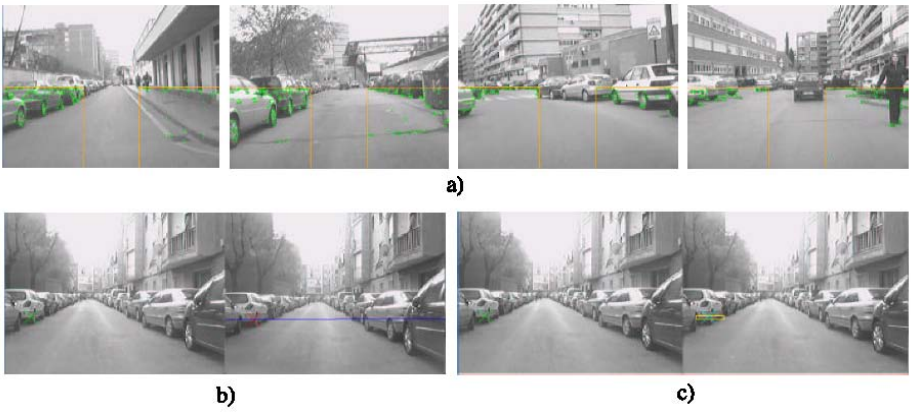


Fig. 1. a) The upper row depicts feature detection results using Harris detector in several images in urban environments. Detection is constrained to a couple of regions of interest located in the lateral areas of the image bellow the horizon line. b) The bottom left image shows an example of features matching in a stereo image. c) The bottom right image depicts an example of feature tracking in two consecutive frames. ZMNCC and mutual consistency check is used both for feature detection and feature tracking.

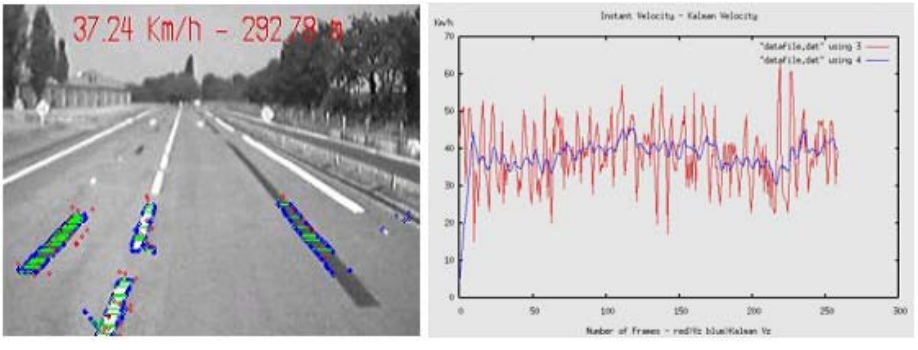
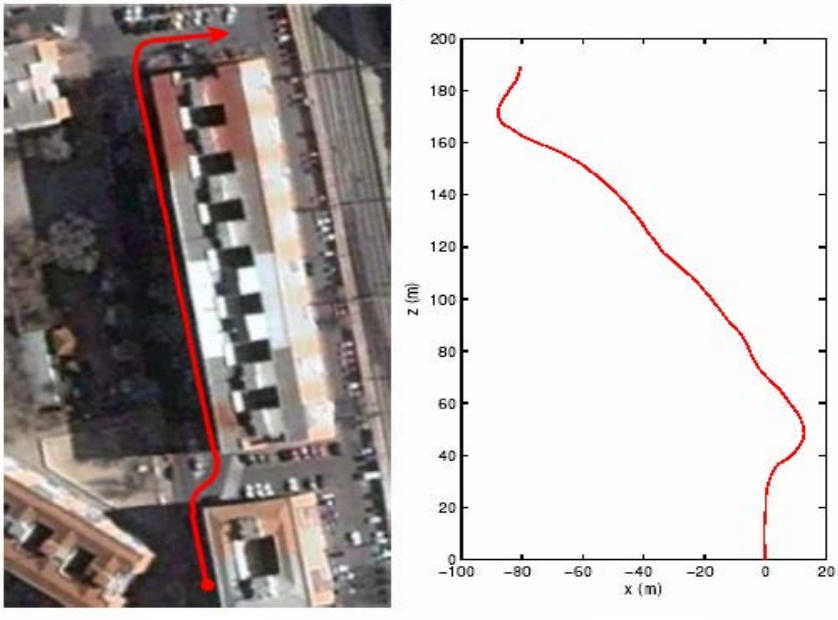


Fig. 2. Detection of coupled road points (left); velocity estimation using vision (right)

example of trajectory estimation using visual odometry. As can be observed, the system provides reliable estimations of the path run by the vehicle in almost straight sections. As a matter of fact, in the experiment the car started turning slight right and then left to run along an almost straight path for a while. After that, a sharp right turn is executed. Then the vehicle moves straight for some metres until the end of the street. Figure 3 illustrates the real trajectory described by the vehicle (a) and the estimated trajectory estimated by the visual odometry algorithm (b). In this case, the estimated trajectory reflects quite well the exact shape and length of the real trajectory executed by the vehicle.



a) b)

Fig. 3. a) Aerial view of the area of the city where the experiment was conducted. b) Estimated trajectory using visual odometry.

5 Conclusions

We have described a method for estimating the vehicle global position in a network of roads by means of visual odometry. To do so, the ego-motion of the vehicle relative to the road is computed using a stereo-vision system mounted next to the rear view mirror of the car. Feature points are matched between pairs of frames and linked into 3D trajectories. The resolution of the equations of the system at each frame is carried out under the non-linear, photogrametric approach using least squares and RANSAC. This iterative technique enables the formulation of a robust method that can ignore large numbers of outliers as encountered in real traffic scenes. The resulting method is defined as visual odometry and can be used in conjunction with other sensors, such as GPS, to produce accurate estimates of the vehicle global position. As part of our future work we envision to develop a method for discriminating stationary points from those which are moving in the scene. Moving points can correspond to pedestrians or other vehicle circulating in the same area. Vehicle motion estimation will mainly rely on stationary points. The system can benefit from other vision-based applications currently under development and refinement in our lab, such as pedestrian detection and ACC (based on vehicle detection). The output of these systems can guide the search for really stationary points in the 3D scene. The obvious application of the method is to provide on-board driver assistance in navigation tasks,

or to provide a means for autonomously navigating a vehicle. For this purpose, fusion of GPS and vision data will be accomplished.

Acknowledgments. This work has been funded by Research Project CICYT DPI2005-07980-C03-02 (*Ministerio de Educación y Ciencia, Spain*).

References

1. Stein, G.P., Mano, O., Shashua, A.: A robust method for computing vehicle ego-motion. In: Proceeding of the IEEE Intelligent Vehicles Symposium, Parma, Italy (2004)
2. Horn, B.K., Weldon, E.J.: Direct methods for recovering motion. International Journal of Computer Vision 2, 51–76 (1988)
3. Meer, P., Mintz, D., Kim, D., Rosenfeld, A.: Robust regression methods for computer vision: A review. International Journal of Computer Vision 6(1), 59–70 (1991)
4. Huber, J., Graefe, V.: Motion stereo for mobile robots. IEEE Transactions on Industrial Electronics 41(4), 378–383 (1994)
5. Zhang, A., Faugeras, O.D.: Estimations of displacements from two 3-d frames obtained from stereo. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(12) (1992)
6. Nister, D., Narodistsky, O., Beren, J.: Visual odometry. In: IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Los Alamitos (2004)
7. Hagnelius, A.: Visual odometry. In: Master Thesis in Computing Science, UMEA University (April 2005)
8. García-García, R.G., Sotelo, M.A., Parra, I., Fernández, D., Gavilán, M.: 3D Visual Odometry for GPS Navigation Assistance. In: IEEE IV Symposium, Istanbul, Turkey (2007)

PSPRT: A Case of Pervasive System for Public Road Transport

Carmelo García, Francisco Alayón, Joaquín Caraballo, Ricardo Pérez, and Gabino Padrón

Universidad de Las Palmas de Gran Canaria, Dpto. Informática y Sistemas, Edificio de Informática y Matemáticas, Campus Universitario de Tafira, 35017 Las Palmas
{rgarcia, falayon, rperez, gpadron}@dis.ulpgc.es

Abstract. We describe a case of pervasive system which is working in a public road transport company of passengers, the description is based on the main elements of ubiquitous system architecture; these are: pervasive devices, the pervasive network and the system middleware. From the point of view of the pervasive devices used, the more important elements is the onboard system installed in the fleet vehicles, it is an embedded computer that is able to achieve all the needed functionalities integrating all the onboard elements (sensors, location devices, payment devices and communications devices) and handling several communications infrastructures in a suitable way by the system middleware it permit to the transport corporation use a quality information of its productions operation executed by the vehicles fleet.

Keywords: Cyber cars, Intelligent Transport Systems and Pervasive Computing.

1 Introduction

Thanks to advances in information technologies, particularly in mobile communications and computing, miniaturization of devices and location systems, the paradigm of ubiquitous computing conceptualized by Mak Weiser at 1991, in his vision about the computer of the XXI century [1], the pervasive model is not just a theoretical model, it is a successfully model applied to developed intelligent environments like: homes [2], teaching [3], tourism [4], etc. However, this model has not been used extensively in the corporative productive systems. This paper describes how the pervasive computing paradigm has been applied successfully to improve the production system of a public transport company, resolving classic problems and incorporating new functionalities. Based on this model, we have designed PSPRT, a system which integrates the mobile information systems installed on the vehicles fleet, into the corporation information system, improving the quality of the services in several aspects such us: fulfilment of timetables, driving safety, payment systems, fleet maintenance, planning development, etc. The main aspects of the system architecture are explained in this article. The more important elements of this architecture are: the onboard embedded system, the pervasive network and the middleware. Nowadays, the system is used by Global Salcai-Utinsa S.A., a road passenger's public transport company which operates in Gran Canaria (Canary Islands, Spain). This company has a fleet of 307 buses, transporting more than 30.000.000 passengers yearly. Our system handles all the information generated by the production activity of this corporation.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1126 – 1133, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Objectives of the System. Why to Apply the Ubiquitous Model?

Traditionally, the transport companies by road have to face a problem that consists of the weak integration of the mobile systems into the corporative information system. This weak integration produces inefficiency in the most critical activity of the company: the productivity. This problem implies difficulties to control the activity of the production resources (drivers and vehicles) and as a consequence, the company can not resolve incidents that can affect to the planning fulfilment, for example, time fulfilment in bus stops. Another consequence of this problem is that the transport enterprise can not automatically execute the data flow between the mobile systems and different nodes of the corporative network. This data flow is needed for important activities such us: payment recovery, user information and maintenance. Considering the principle of the ubiquitous paradigm: "connectivity ever and every place", we think that the methods of this paradigm can solve this problem. The pervasive model is oriented to the interoperability and scalability of the systems. This is another reason to apply this model to the road transportation. We can resume the main objectives reached by our system in the following way:

- Improving the quality of the company service using mobile systems, more integrated into the corporative information system. This best integration implies a better knowledge of the fleet operational state and this way, a better capacity to act before events affect the planning..
- Improving the security and comfort of drivers and passengers. The system on board helps the driver in his entire task, reducing to the maximum its interventions and distractions.

3 Description of the System

The ubiquitous system that we have developed permits the systems on board in vehicles of the fleet of a transport company to be seen like another node of the corporative network. This integration has allowed automation of very important tasks, and the adding of new functionalities. PSPRT improves the following components of the transport company information system:

- Fleet operational control subsystem: It monitors and takes the necessary actions in order to ensure the fleet's operating plan fulfilment. It is based on a control model managed by events, avoiding the need of polling when looking for changes at the environment.
- Information to client subsystem: Its task is to give the most current information about the services offered by the company.
- Production information subsystem: It is responsible of managing all the information generated in the vehicles as a result of clients using the services. All information flow is executed in automatic way, from the client to the corporative system, avoiding the lost of important data.
- Fleet maintenance subsystem: This element is responsible of to grant the correct operation of hardware devices and software on board. With our system the company has developed tools that facilitate this work.

Any of those mentioned subsystems is configured by a set of pervasive applications that run either in a mobile or in a fixed system and using different communications infrastructures. To explain the most relevant aspect of PSPRT we will base on the components of a ubiquitous architecture according to Saha [5].

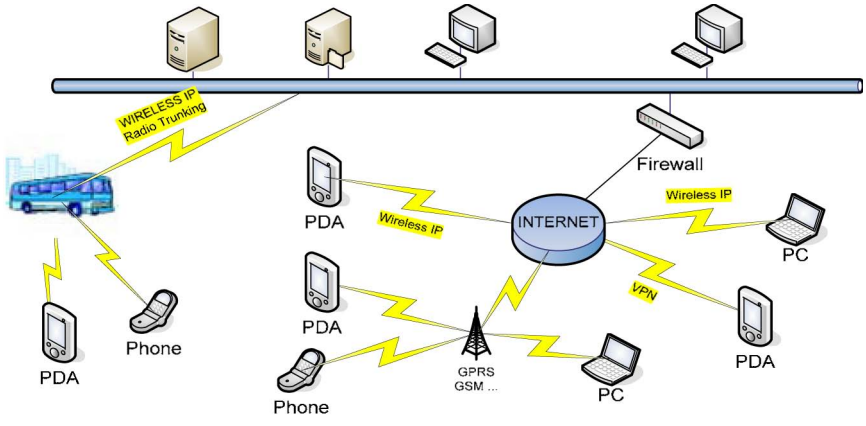


Fig. 1. A General vision of the system. The system uses several kind of communication infrastructures, permitting to see the on board system like another node of the corporation network.

3.1 Devices

To obtain a high degree of accessibility, system users with their different roles can use a wide variety of devices. Specifically, users can retrieve available information by means of PDA's, mobile telephones, or PC's. Clients can use prepay cards, cards without contact, and mobile phones. Maintenance, operation control personnel and inspectors can use PDA's and computers to access to the information that they require. The most important element is the computer carried aboard. This is an embedded system qualified to operate at special conditions (vibrations, irregular variations of temperature and electrical disturbances). This computer has all the necessary resources to support all the functionalities that will be likely required by systems aboard. It has sensors for surveying the electrical conditions and temperature levels of the vehicle. It also has a presence sensor to report about the fuel level. Some vehicles have video cameras to record the activities inside the vehicle for security reasons. In order to know the position of the vehicles and to synchronistic the fleet operation the on board system uses a GPS receiver. The embedded on board computer has a variety of network interfaces: wired and wireless connecting devices between them and to the computer on board, and the computer on board to external systems.

3.2 Ubiquitous Network

The communications network in a ubiquitous system is responsible for data interchange between mobile and fixed devices integrated into the system. In order to play this role, our network needs to use infrastructures of mobile telephony and radio trunking for wide range communications, wireless local networks based on

IEEE802.11 for short range communications that require high bandwidth, and, for very short range communications, it uses Bluetooth and infrared technology. Another relevant aspect of ubiquitous networks is the use of normalized networks like Internet —as context awareness requires a high degree of accessibility— by a variety of mobile devices like PDA's, mobile phones, portable PC's. Ad-hoc networks are also used when a high degree of security is needed or we want to communicate with devices with specific characteristics. In this network real time communications are very important in order to inform to the control centre about the accomplishment of the fleet planning. Next, we describe the different infrastructures of communication that are being used by the different subsystems:

- Radio trunking: This is used to communicate data packets related to events and alarms that are produced during normal operations on mobile systems. This infrastructure has a high availability for mobile communications at long distances but has a much reduced bandwidth.
- Wireless communications: Based on IEEE 802.11, allow the interchange of great amounts of data between the control centre and systems on board. Due to its properties of high bandwidth and low cost, it is used by all subsystems to communicate with systems on board. We have developed communication protocols in order to obtain spontaneous integration between mobile stations into the corporative system and the transmission with minimal errors rate in spite of the speed of the vehicles.
- Mobile telephony: It is used to interchange information with clients, for example, to provide them services information; or to interchange information with the operation control personnel.
- Very short range communication: This infrastructure is used to transfer data packets of small sizes that allow executing in automatic way some tasks like fuel level control, automatic payment (cards without contact). This system is based on very low range RF that uses ad-hoc protocols which main characteristics consist of maintaining the security of transactions.

PSPRT has a set of protocols allowing two different communication models. One model permits the applications to interchange data by dynamic channels. This model is used to transfer priority messages over long, short and very short distances. The other model is based on mailboxes or buffers where non-priority messages are placed to be transmitted later over long and short distances. This scheme gives us the following advantages: first, it is uniform and transparent for the applications, and second, it can be integrated into more complex models of communications. According to Cao [6], protocols based on mailboxes in mobile environments can be characterized by three basic properties:

- The degree of migration of mailboxes, referred to the capacity of the system to permit the migration of mailboxes that are associated to applications. Three possibilities are distinguished: one, static mailboxes, located ever on the same station, two, mailboxes that move along with applications that are related to, three, independent mailboxes, that can be moved independently of the application to which it is associated.
- Delivery of messages. Referred to how the messages are given to its targets. We distinguish two possibilities: one, named *push* operations that need to know every

moment where the application targets are, two, named *pop* operations, where the messages are given to application only when these ask for them.

- Synchronization between mailboxes and delivery of messages. Referred to the transfer of message between mailbox and application with minimum error rate. We distinguished the following options: one, when mailboxes don't move it is not necessary the synchronization, two, synchronization between host and mailbox that has moved on, three, synchronization between the mailbox and the application that has moved on with respect to mailbox, four, a combination of two previous cases.

Concentrating us on our protocol, we don't allow migration of mailboxes, because of it is the more suitable option for small and medium size networks. Our network delivers the messages using the *push* method, which is the most suitable in a real time context. Therefore, we don't need to synchronize mailboxes and delivery of messages. When a mobile application connects to a static station, it sends a registry message and waits for an acknowledge message, then the message can be sent to the application. For every message received by the mobile application, a reception confirmation is sent back.

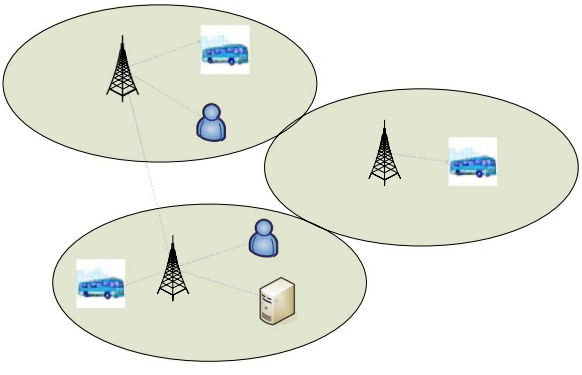


Fig. 2. System communication infrastructures

3.3 Middleware

In a pervasive context, the middleware is the interface between ubiquitous applications and the pervasive network. This element has to provide the following functionalities:

- A transparent environment for communications; the details of the different networks (standard and ad-hoc) used in the pervasive network must be supported by the middleware and not by the applications.
- Context information; the details of how this information context is obtained are a responsibility of the middleware and not of the pervasive applications.
- Resources to specify which actions to execute related to the different contexts where the pervasive applications are executing.
- A uniform development environment to define the context dependences of pervasive applications. The specifications have to be independent of the hardware and the operating system used by the pervasive applications developers.

In general, the middleware consist of two elements: the development environment and the active part or run-time, which permits pervasive applications to communicate transparently and to obtain the needed context information. Next, we describe the main characteristics of the middleware included on our system.

The context modelling. In our model, any context relevant data needed by the applications may be broken up in basic elements named "context entities", or simply entities. Every entity is defined specifying the three attributes: first, the symbolic identification that is the name used by applications to identify the entity in the system, second the data type: that defines the set of values from which the entity may take its value, and finally the system location: which consists of a complete specification of the access path to the data value of the entity. In PSPRT the specification of all entities is structured in a hierarchical way, providing different spaces of information, which are associated to the different functional subsystem, for example: the system context formed by all the entities needed to achieve the automatic data flow, the client context configured by all the entities required by the traveller's information subsystem, the production context: formed by all the entities related to the client's movements and payment recovery, the control context configured by all the entities required to achieve the control operations of the vehicles fleet and , finally, the maintenance context: configured by all the entities related to the administration of the software elements and the maintenance of devices aboard.

The context-triggered actions management. The flow transaction control is achieved by rules that are executed automatically by the different mobile applications. These rules consider technological and corporate strategic aspects which are modelled using context information obtained by the system middleware. For this reason, from the point of view of pervasive applications, PSPRT can be described as a set of applications based on context-triggered rules which produce information and later transmits it to other applications using transparent and generic communication channels. In this model, the middleware run-time provides these channels and it also establishes scheme of communication to be used (direct or using mailboxes). The rules have a common structure configured by three components: the first element is the specification of the set of context entities used by the rule, the second is the logical expression which defines the triggered condition; and the third is the expression is based on a formal grammar. Following, we present a specification of a set of rules.

Example of set of rules

```
# Used resources control section
(driver == 1234) warn 9;
# Maintenance section
(dev[CONSOLE] == ERROR) warn 29 CONSOLE;
(dev[CANCEL] == ERROR)  warn  29 CANCEL;
# Exploitation section
(state & WORKING)       warn 22
(state == FULL)         warn 25
(state == TOO_FAST)     warn 24
(state == TOO_SLOW)     warn 23
```

The spontaneous interaction. In a pervasive context, there are two reasons for spontaneous interaction between applications: first, the movement of the platforms where the applications are executed, which produces spontaneous connections, and second, context changes of the applications. Due to these changes, the applications have to adapt to the new context dynamically. An important requirement to support the spontaneous interaction is that it has to take place in heterogeneous networks and devices. In some cases, this spontaneous interaction is supported by an asymmetric communication scheme based on a client-server model. In this scheme, the applications requests are transmitted over the network and these requests are attended by a network server. In this method, the applications have to find the servers in the network dynamically or, alternatively, identify the server using previous information. An alternative scheme is used in our middleware that uses a symmetrical communication scheme. In this scheme, we establish two kind of symmetrical communications: the first type is used to transmit high priority data, normally related to real time requirements, by means of a direct communication between applications. The second type consists of a scheme based on mailboxes.

4 Results

Nowadays, PSPRT is working in Global Salcai-Utinsa S.A. Company; it is a road passenger's public transport corporation which operates in Gran Canaria (Canary Islands, Spain). All the information produced by the bus fleet activity is handled by PSPRT. To illustrate the amount and the complexity of this management, we can comment some general figures of the company activity: the number of buses of the fleet is 307 PSPRT controls in real time all the operations of the vehicles, all the data flow generated by the 30.000.000 of passengers that this company transports yearly is handled in automatic way by PSPRT. This activity generates more than 180.000 data transactions per day, all of them automatically processed by the system with a lost data ratio bellow of 0.1%.By PSPRT the company can achieve the operation control and technical maintenance of the vehicles fleet, for example: the company knows the produced exceptions in its buses (timetable delays, technical alarm, full vehicle condition, etc) in less than one minute, or the company can update in automatic way the data or program of all the on boarded systems in lee than two days

5 Conclusions

The pervasive paradigm can help us to solve common problem of the intelligent transport system, basically related to the suitable use of the communications infrastructures, the automatic data flow and the accessibility of the users. Thank to this model, we can develop intelligent environment to increase the functionalities and safety of the vehicles. To demonstrate these conclusions we have described a system called PSPRT based on this paradigm. Nowadays this system is working in a public transport company which transports more of 35.000.000 of passenger yearly with a vehicles fleet of 307 buses that realises about 85000 expeditions daily. By PSPRT the company can achieve the operations control of the fleet in real time and the data flow of its production is executed automatically.

References

1. Weiser, M.: The computer for the 21st century. In: IEEE Pervasive computing, mobile and ubiquitous systems, vol. 1(1), pp. 18–25. IEEE Computer Society Press, Los Alamitos (2002) (reprinted with permission Copyright 1991 by Scientific American Inc)
2. Helal, S., et al.: The Gator Tech Smart House: a programmable pervasive space. Computer 38(3), 50–60 (2005)
3. Johanson, B., Fox, A., Winograd, T.: The Interactive Workspaces Project: Experiencies with ubiquitous computing rooms. IEEE Pervasive computing, mobile and ubiquitous systems 1(2), 67–74 (2002)
4. Belloti, F., et al.: A: User Testing a Hypermedia Tour Guide. IEEE Pervasive computing, mobile and ubiquitous systems 1(2), 67–74 (2002)
5. Saha, D., Ukherjee, A.: Pervasive Computing. A paradigm for 21st Century. Computer 36(3) (2003)
6. Cao, et al.: Mailbox-Based Scheme for Mobile Agent Communications. Computer 35(9) (2002)

Study of Correlation Among Several Traffic Parameters Using Evolutionary Algorithms: Traffic Flow, Greenhouse Emissions and Network Occupancy

Javier Sánchez Medina, Manuel Galán Moreno, and Enrique Rubio Royo

Innovation Center for Information Society – C.I.C.E.I.,
University of Las Palmas de Gran Canaria,
Campus de Tafira s/n, Las Palmas de Gran Canaria, Spain
jsanchez@polaris.ulpgc.es, mgalan@cicei.com, rubio@cicei.com
http://www.cicei.com*

Abstract. During the last two years we have been working on the optimisation of traffic lights cycles. We designed an evolutionary, distributed architecture to do this. This architecture includes a Genetic Algorithm for the optimisation. So far we have performed a single criterion optimisation – the total volume of vehicles that left the network once the simulation finishes. Our aim is to extend our architecture towards a multicriteria optimisation. We are considering Network Occupancy and Greenhouse Emissions as suitable candidates for our purpose. Throughout this work we will share a statistical based study about the two new criteria that will help us to decide whether to include them or not in the fitness function of our system. To do so we have used data from two real world traffic networks.

Keywords: Genetic Algorithms, Cellular Automata, Traffic Optimisation, Greenhouse Emission, Multicriteria Optimisation.

1 Introduction

The traffic issue means, for every major city in the world, not only a quality and comfort factor, but also a very important economic subject.

The non-stopping overload process that traffic infrastructures are suffering calls for new solutions. Moreover, the world governments are aware about the global warming risks. Hence, it is a must to optimise the existing infrastructures to obtain the very best service they can provide, minimising their environmental impact.

In our group we have been working on the optimisation of traffic lights cycles for the better performance of urban traffic networks. As shown in [1], traffic light

* This research have been partially supported by the Ministerio de Educación y Ciencia of Spain (ref TSI2004-05949), which is financed 70% from European Union FEDER funds.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1134–1141, 2007.
© Springer-Verlag Berlin Heidelberg 2007

cycles have a strong influence in traffic flow results. We have designed a three pillar model consisting of a Genetic Algorithm (GA) as optimisation technique, a traffic microscopic simulator inside the fitness function of the GA, and a Beowulf Cluster as scalable MIMD multicomputer.

We have been testing this architecture through two years with successful results. So far we have used a single criterion for the fitness function of the Genetic Algorithm – the total number of vehicles that left the network once the simulation finishes.

Currently, our aim is to extend this system including other criteria. We are now focused on the occupancy of the network and the greenhouse emissions as new criteria candidates. Across this paper we present a statistical study comparing the three criteria, and using two real world networks as test cases. The rest of the article is organised as follows. In the next section we describe our methodology. In section 3 we present the new criteria considered. In section 4 we show the tests performed and the obtained results. Finally, some concluding remarks and future work ideas are presented in section 5.

2 Methodology Description

The architecture of our system comprises three items, namely a Genetic Algorithm (GA) as Non-Deterministic Optimisation Technique, a Cellular Automata (CA) based Traffic Simulator inside the evaluation routine of the GA, and a Beowulf Cluster as MIMD multicomputer. Through this section we will give a succinct description for the GA and the CA Traffic Simulator used. Finally, a brief description of the Beowulf Cluster will also be given.

2.1 Genetic Algorithm

- *Fitness Function.* After testing several criteria we found out that we obtained the best results by using the absolute number of vehicles that left the traffic network during the simulation as criterion.
- *Chromosome Encoding.* A Gray Code encoding ([2]) is used to codify the length (seconds) of every one of the states associated to an intersection. The states sequence is preestablished.
- *Selection Strategy.* We have chosen a Truncation and Elitism combination as selection strategy. This combination seems to be the most fitted to our problem among a set of selection tested strategies.
- *Crossover Operator.* We use a standard Two Points Crossover.
- *Mutation Operator.* The value of a randomly chosen bit in the chromosome is just flipped. The mutation probability is variable, starting with a very high probability that decreases generation by generation.

2.2 Traffic Simulator

Inspired on the Cellular Automata Model we have developed a non-linear model for simulating traffic behaviour. We have developed a traffic model based on the

SK[1] model ([3]) and the SchCh[2] model ([4]). The SchCh model is a combination of a highway traffic model — Nagel-Schreckenberg ([5]) — and a very simple city traffic model — Biham-Middleton-Levine ([6]). The SK model adds the "smooth braking" for avoiding abrupt speed changes. We decided to modify our model inspired by the SK model due to its improved results for all the tests shown in Brockfeld et al. ([7]).

2.3 Beowulf Cluster

The Architecture of our system is based on a five node Beowulf Cluster, due to its very interesting price/performance relationship and the possibility of running Open Software on it. On the other hand, this is a very scalable MIMD computer, a very desirable feature in order to solve all sort — and scales — of traffic problems.

3 Optimisation Criteria Under Study

3.1 Traffic Flow

As explained in section 2, in our methodology we use the total volume of vehicles that left the network once the traffic simulation finishes as fitness value. In order to compare that parameter with the Total Emissions and Occupancy relative parameters – subsections 3.2 and 3.3 – it is required to derived a new relative parameter. So, for this current study we define the Exit Probability (equation 1). The Exit Probability is the number of vehicles that left – N_v^l – over the number of vehicles that entered the network – N_v^i – once the simulation finishes.

$$\text{Exit Probability} = \frac{N_v^l}{N_v^i} \qquad (1) \qquad TE = \sum_{c=0}^{N_{cells}-1} (f_{EF}(i,c)) \qquad (2)$$

3.2 Total Emissions Criterion

In [8] it is shown how CO and NO_x are mostly linearly correlated with the speed of vehicles. This fact may be clearly observed in pictures 4 and 5 of that work. With this in mind we define a new parameter, Emission Factor, as the scalar value of the speed of every vehicle, every simulation iteration. This such low computing consuming parameter will give us an approximated idea of the volume of greenhouse gases *emitted* during the simulation.

In our tests we registered iteration by iteration the global sum of emissions – Total Emission (TE) – as shown in equation 2. In this equation N_{cells} means the number of cells in the traffic network; and f_{EF} means the Emission Factor of the vehicle at cell 'c' and iteration 'i'. Obviously, when the cell is not occupied, the f_{EF} value is set to 0.

We believe that the minimisation of TE would mean the minimisation of emissions, and we want to test if it is worth to be included in the fitness function

[1] Stephan Krauss, the author.
[2] Andreas Schadschneider and Debashish Chowdhury, the authors.

or not. Throughout this study we have used the Mean and Standard Deviation of TE calculated across all the simulation iterations.

3.3 Occupancy Criterion

The second criterion we have considered is the Occupancy. During the optimisation we obtain the average "State of Congestion"(SOC) and the average "Time of Occupancy" (TOC). SOC was defined in [9]. TOC was defined in [10]. In equations 3 and 4 we represent both parameters. About equation 3, N_c^o means the number of cells occupied by a vehicle, and N_c^T means the total number of cells in the treated network. About equation 4, N_{it}^o means the number of simulation iterations that a particular cell is occupied by any vehicle, and N_{it}^T means the whole number of iterations that the simulation lasts.

As one may infer from equations 5 and 6, the average SOC across all the simulation iterations and the average TOC across all the cells in the network are the same value. In other words, the mean value of the average occupied cell ratio across all the simulation iterations is equivalent to the mean value of the average number of occupied iterations for a particular cell across all the cells in the traffic network.

$$SOC = \frac{N_c^o}{N_c^T} \quad (3) \qquad TOC = \frac{N_{it}^o}{N_{it}^T} \quad (4)$$

$$\overline{SOC} = \frac{\sum_{i=0}^{N_{it}^T} \frac{N_c^o(i)}{N_c^T}}{N_{it}^T} = \frac{\sum_{i=0}^{N_{it}^T} N_c^o(i)}{N_c^T * N_{it}^T} \quad (5)$$

$$\overline{TOC} = \frac{\sum_{c=0}^{N_c^T} \frac{N_{it}^o(c)}{N_{it}^T}}{N_c^T} = \frac{\sum_{c=0}^{N_c^T} N_{it}^o(c)}{N_c^T * N_{it}^T} \quad (6)$$

In this research, we have considered two different moments for the Occupancy. They are the average and the standard deviation of SOC values measured at every iteration across the whole simulation.

4 Statistical Study

4.1 Test Results

We have used two traffic networks for this research – Figures 1 and 2. The first one is situated in "Las Ramblas" – Santa Cruz de Tenerife, Canary Islands, Spain. The second one is situated in "La Almozara", in the city of Zaragoza, Spain. The size of the network was 1643 cells for "Las Ramblas" and 2753 cells for "La Almozara".

We have used a population of 200 individuals, and we have run the Genetic Algorithm through 200 generations. For each test case we have run 30 executions of the Genetic Algorithm. The traffic simulations run during 2000 iterations for "Las Ramblas" and 4000 iterations for "La Almozara".

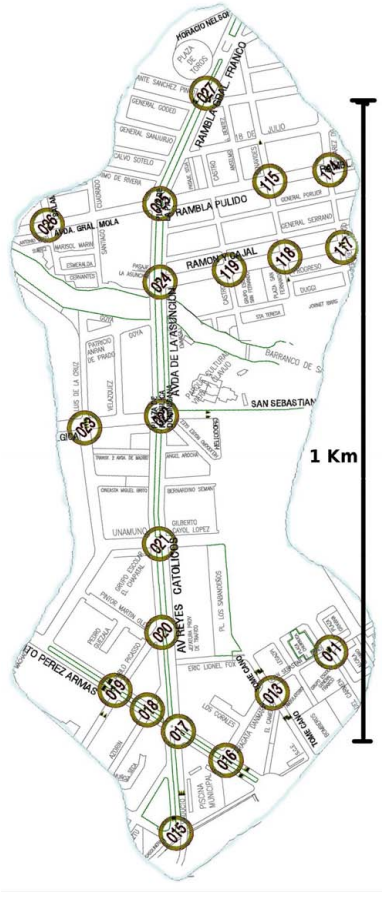


Fig. 1. "Las Ramblas" zone

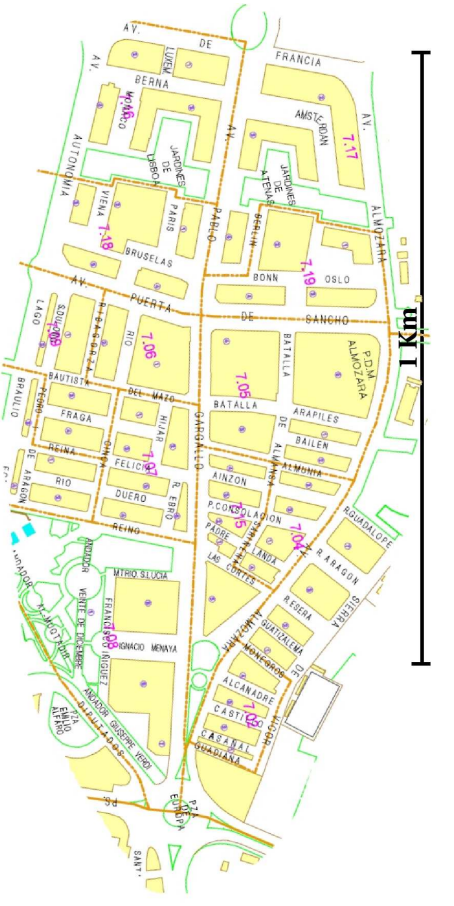


Fig. 2. "La Almozara" zone

Once the 60 executions of the GA finished we picked up the following parameters from the final populations: Total Flow (Fitness), Occupancy and Total Emissions. Our aim is to see how correlated are they. As we said in 3.3, every iteration the average and the standard deviation of SOC were measured. We sampled the average and standard deviation of Total Emissions too.

In figures 3, 4, 5, 6, 7 and 8 we have represented 4 pairs of parameters for "Las Ramblas" and 2 pairs for "La Almozara". In these pictures we have also represented the linear regression function using the LMS fitting algorithm at each subplot. For that purpose we have used the *polyfit* and *polyval* functions of Matlab. Finally, we have also calculated the Pearson Correlation Coefficient (equation 7) for each case.

$$\rho = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{7}$$

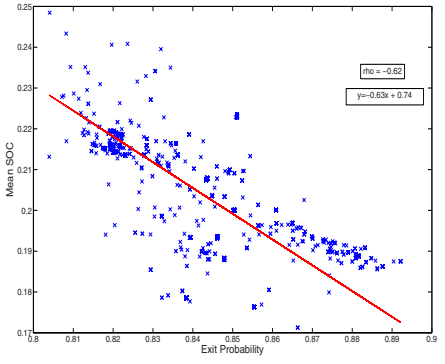


Fig. 3. Las Ramblas – Exit Probability Versus Mean SOC

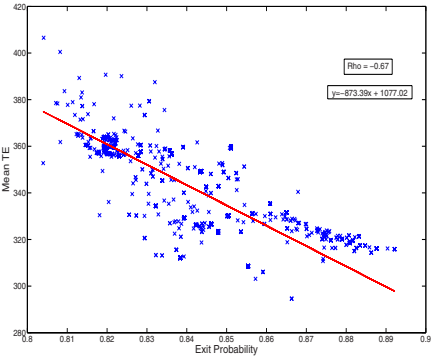


Fig. 4. Las Ramblas – Exit Probability Versus Mean Total Emissions

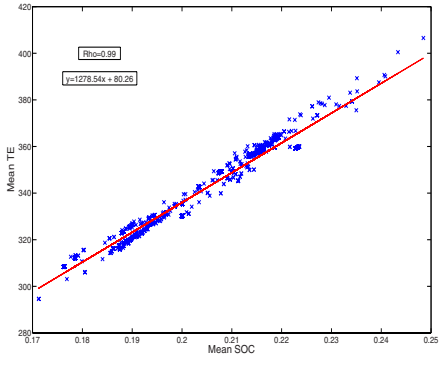


Fig. 5. Las Ramblas – Mean SOC Versus Mean Total Emissions

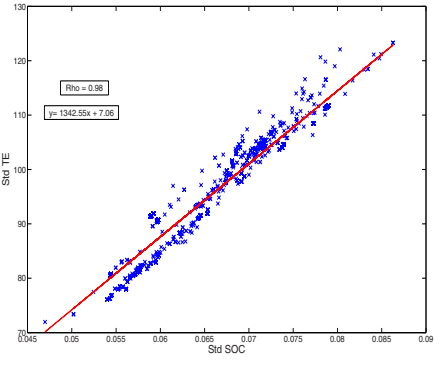


Fig. 6. Las Ramblas – Std. SOC Versus Std. Total Emissions

4.2 Discussion

First of all, one may see that there exists a strong linear relationship between the mean and std. SOC and the mean and std. Total Emissions respectively – figures 5,6, 7 and 8. This fact is specially clear in figure 7, where the Pearson coefficient is 0.99.

It seems clear that the Occupancy and Total Emissions criteria are very related. This fact conforms with reality since a very loaded traffic network is also likely to be a very pollutant one, and vice versa.

Hence, we can not use the two criteria simultaneously in a hypothetical multicriteria fitness function. We can either minimise the emissions while maximising the occupancy or the reverse. Thus, we will have to choose between the minimisation of emissions or the maximisation of the occupancy.

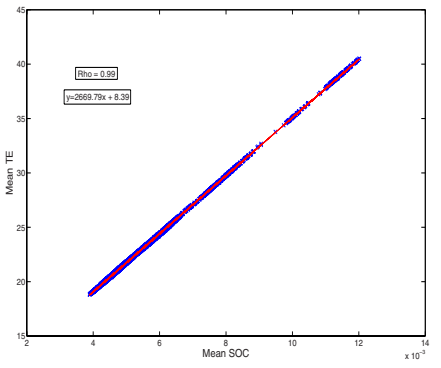


Fig. 7. La Almozara – Mean SOC Versus Mean Total Emissions

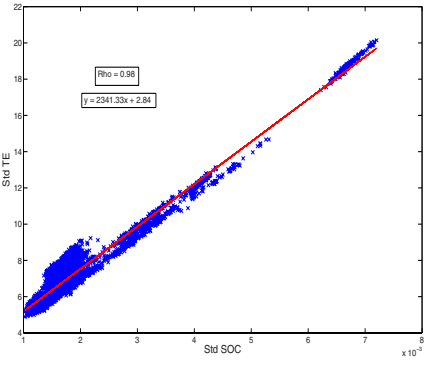


Fig. 8. La Almozara – Std. SOC Versus Std. Total Emissions

Secondly, as may be observed in figure 3, there exist a clear relationship between the Exit Probability and mean SOC. This is not a linear relationship, but a higher order one. In further research we will explore this.

Also from figure 3 is observed that the Exit Probability and mean SOC are negatively associated. This is consistent with common sense, since a congested network is likely to have a small Exit Probability values and vice versa.

Figure 4 shows another case of nonlinear negatively associated relationship between the Exit Probability and Emissions. Again, this makes sense since experience says that a very pollutant network – for instance a congested network – is likely to have a low Exit Probability value.

5 Conclusions

Throughout this paper we have presented a study comparing several parameters for developing a multicriteria fitness function, extending the consolidated traffic light optimisation methodology we designed in our group.

In this work we have tested several criteria related to network occupancy and greenhouse gasses emissions using two real world traffic networks: "Las Ramblas" in Santa Cruz de Tenerife, and "La Almozara" in Zaragoza – Spain.

The results are enlightening and may be summarised as follows. First we have found a strong linear and positive correlation between the occupancy of the network and the emissions criteria. This is an important new information. Because of it, in a hypothetical multicriteria fitness function we will have to choose between maximising the occupancy of the network or minimising the emissions.

We have observed nonlinear relationships between occupancy and the Exit Probability too. A similar effect was observed when comparing the Exit Probability and Total Emissions.

For future work we plan to determine the degree of the nonlinear relationships found in this research. We are also planning to test new criteria. Once chosen suitable criteria we will be able to define a new multicriteria fitness function for our methodology.

Acknowledgements

The authors would like to acknowledge the Santa Cruz de Tenerife and Zaragoza City Council Traffic Departments for their kind help. We specially would like to thank Mr. Hilario Rodríguez González from Santa Cruz de Tenerife for his willingness to collaborate with us, making this work feasible.

We also would like to thank Professor María Luisa Sein-Echaluce Lacleta, from the University of Zaragoza for her kind help to obtain the data from the Zaragoza Traffic Department.

References

1. Brockfeld, E., Barlovic, R., Schadschneider, A., Schreckenberg, M.: Optimizing Traffic Lights in a Cellular Automaton Model for City Traffic (2001), http://arxiv.org/ps/cond-mat/0107056
2. Black, P.E.: "Gray Code", from Dictionary of Algorithms and Data Structures. In: Black, P.E. (ed.) NIST, http://www.nist.gov/dads/HTML/graycode.html
3. Krauss, S., Wagner, P., Gawron, C.: Metastable states in a microscopic model of traffic flow. Phys. Rev. E 55, 5597–5605 (1997)
4. Schadschneider, A., Chowdhury, D., Brockfeld, E., Klauck, K., Santen, L., Zittartz, J.: A new cellular automata model for city traffic. In: Traffic and Granular Flow 1999: Social, Traffic, and Granular Dynamics, Springer, Berlin (1999)
5. Nagel, K., Schreckenberg, M.: A Cellular Automaton Model for Freeway Traffic. Journal de Physique I France 33(2), 2221–2229 (1992)
6. Biham, O., Middleton, A.A., Levine, D.: Phys. Rev. A 46, 6124 (1992)
7. Brockfeld, E., Khne, R.D., Wagner, P.: Towards Benchmarking Microscopic Traffic Flow Models Networks for Mobility. In: International Symposium, vol. I, pp. 321–331 (2002)
8. Kean, A.J., Harley, R.A., Kendall, G.R.: Effects of Vehicle Speed and Engine Load on Motor Vehicle Emissions. Environmental Science and Technology 37(17), 3739–3746 (2003)
9. Bham, G.H., Benekohal, R.F.: A high fidelity traffic simulation model based on cellular automata and car-following concepts. In: Transportation Research Part C, vol. 12, pp. 1–32. Elsevier, Amsterdam (2004)
10. May, A.D.: Traffic Flow Fundamentals, vol. 78. Prentice-Hall, Englewood Cliffs (1990)

Robust Obstacle Detection Based on Dense Disparity Maps

Wided Miled[1,2], Jean Christophe Pesquet[2], and Michel Parent[1]

[1] INRIA, IMARA Team, Domaine de Voluceau,
78150 Le Chesnay Cedex, France
{wided.miled,michel.parent}@inria.fr
[2] Institut Gaspard Monge / UMR-CNRS 8049,
Université Marne-la-Vallée, 77454 Champs-sur-Marne, France
pesquet@univ-mlv.fr

Abstract. Obstacle detection is an important component for many autonomous vehicle navigation systems. Several methods have been proposed using various active sensors such as radar, sonar and laser range finders. Vision based techniques have the advantage of relatively low cost and provide a large amount of information about the environment around an intelligent vehicle. This paper deals with the development of an accurate and efficient vision based obstacle detection method that relies on dense disparity estimation between a pair of stereo images. Firstly, the problem of disparity estimation is formulated as that of minimizing a quadratic objective function under various convex constraints arising from prior knowledge. Then, the resulting convex optimization problem is solved via a parallel block iterative algorithm which can be efficiently implemented on parallel computing architectures. Finally, we detect obstacles from the computed depth map by performing an object segmentation based on a surface orientation criterion.

Keywords: Obstacle detection, stereo vision, convex optimization, depth estimation, object classification.

1 Introduction

The ability to detect obstacles on the road is essential for intelligent vehicle navigation systems. A large variety of active and passive sensors have been used to perform such a task. Active sensors like radars or laser range finders have been shown to provide good performance and have the advantage of directly giving distance measurements. However, they are expensive and are very sensitive to weather conditions. Moreover, their lower spatial resolution does not enable the detection of small obstacles.

On the other hand, passive sensors like monocular or stereo vision [1], [2], [3], [4], [5] provide a large amount of information about the environment around the vehicle at a relatively low sensor cost. However, obstacle detection techniques involving vision are generally computationally expensive. Cooperative approaches

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1142–1150, 2007.
© Springer-Verlag Berlin Heidelberg 2007

using fusion between both type of sensors take advantage of their complementary features while leading to improved detection results [6].

Vision based approaches for obstacle detection can be classified into two main categories. Monocular approaches, from a single camera, assume some *a priori* knowledge about the obstacles to be detected and use techniques such as color segmentation, or detection of specific characteristics such as texture or symmetry axes [2]. The computation of the optical flow from road image sequences can also be used [4], [7]. Most of the flow based methods perform a segmentation process of the flow, in which objects having a different motion from the road motion are detected as obstacles. While accurate results have been obtained, these approaches fail when the obstacles have small or null speed.

Stereo vision techniques enable recovering the depth information of a scene from a pair of left and right views. They involve finding corresponding pixels in both images, leading to the so-called disparity map. Most of stereo vision algorithms use a ground plane estimation to distinguish obstacles in the disparity image from the road surface. A robust and real-time method for doing this was developed by Labayrade et al. [1]. In this approach, obstacles are detected in a V-disparity space, which is an histogram of the disparities on every line of the disparity image. According to their notations [1], V corresponds to the vertical coordinate in the (U,V) image coordinate system. In the V-disparity domain, the road surface profile is represented by line features, extracted by a Hough transform and obstacles in the vertical plane are projected as vertical lines. Hu et al. [8] extend Labayrade's work by proposing an U-V disparity concept to classify the 3D scene points into ground surface, road side surfaces and obstacles.

In this paper, we describe a new strategy to detect obstacles from a stereovision system mounted on a vehicle. Our method works directly in the disparity image space and exploits all its relevant information to detect obstacles. To deal with illumination variations often encountered in practice, we develop a spatially varying multiplicative model that accounts for brightness changes between both images in the stereo pair. The stereo matching problem based on this model is then formulated as a constrained optimization problem in which an appropriate objective function is minimized under convex constraints. These constraints arise from prior knowledge and rely on various properties of the fields to be estimated. Object segmentation based on the depth map, makes use of a surface orientation criterion, and enables reliable and robust object detection.

The paper is organized as follows: section 2 briefly describes the stereo vision configuration, section 3 presents a robust approach aimed at the estimation of the disparity map. Section 4 describes the obstacle segmentation procedure and provides some experimental results. Finally, a conclusion is given in section 6.

2 Stereo Vision Configuration

The obstacle detection system proposed in this paper will be integrated on the Cybercar platform which is a road vehicle with fully automated driving

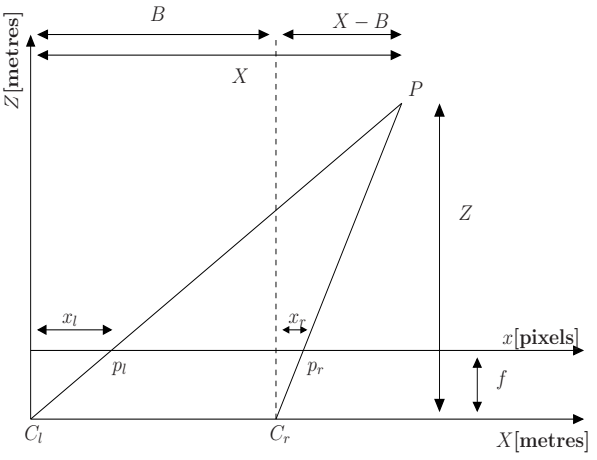


Fig. 1. Relationship of disparity and depth

capabilities [9]. The video acquisition system adopted on the this platform is the Bumblebee stereo vision system from Point Grey Research. It contains two pre-calibrated digital CCD cameras in an epipolar configuration. This configuration is usually achieved in practice by rectifying the stereo images based on intrinsic and extrinsic parameters of the cameras [10]. The epipolar geometry constraint reduces the correspondence problem from two dimensions to one, by involving that corresponding pairs of image elements always occur on homologous epipolar lines. In this case, the disparity u is calculated as the horizontal displacement between the reference pixel in the left image and the candidate pixel in the right image. Given two corresponding pixels $p_l = (x_l, y_l)$ and $p_r = (x_r, y_r)$ in the left and right images, respectively, the disparity is given by $u = x_l - x_r$. The depth information can then be calculated by triangulation as shown in Figure 1. From similar triangles we derive that:

$$x_l = f\ \frac{X}{Z} \quad \text{and} \quad x_r = f\ \frac{X-B}{Z}, \tag{1}$$

and finally derive

$$Z = \frac{B\ f}{u}, \tag{2}$$

where B is the length of the baseline and f is the camera's focal length. From equation (2) we conclude that disparity is inversely proportional to depth. A disparity map that records the disparity for each image point is therefore sufficient for a complete 3D reconstruction of the scene. Due to the poor quality of the 3D data obtained from the Bumblebee stereo vision system, we need to perform our disparity map estimation that will be described in the next section.

3 Dense Disparity Map Estimation

In order to compute a three-dimensional scene reconstruction, we need to identify, in both left and right images, points that are the projection of the same 3D point on the two image planes. Many matching approaches have been proposed including feature-based and area-based approaches. The feature-based approach provides accurate information and has a low computational cost. However, methods computing only sparse depth information will generally fail to perform well an object segmentation step. Area based methods on the other hand produce dense disparity maps, but often fail at depth discontinuities and in textureless areas. Another difficulty arises from lighting variations between both images, which may occur in outdoor environment and can largely degrade the performance of the depth recovery.

In this paper, we propose a new dense disparity field estimation approach that avoids many of the limitations mentioned above. In what follows, we will briefly describe this approach, due to space limitation, but the reader is referred to [11] for complementary details.

3.1 Stereo Formulation

The standard mathematical model used to compute the disparity map is based on the constant brightness assumption : two homologous points in the left and right views have identical intensity values. To take into account possible illumination change, we replace this assumption with a more realistic model that is close to the one used in [12]. The intensities of two corresponding points are then related by a spatially varying multiplicative term:

$$I_r(x - u(s), y) = v(s)\ I_l(s)\ , \tag{3}$$

where I_l and I_r are the left and right images of a stereo pair, respectively, $s = (x, y)$ is a spatial position in either image, u denotes the horizontal disparity and v represents the multiplicative coefficient of the intensity illumination change. Based on this model, we can compute u and v by minimizing the following cost function based on the Sum of Squared Differences (SSD) metric:

$$\widetilde{J}(u, v) = \sum_{s \in \mathcal{D}} \left[v(s)\ I_l(s) - I_r(x - u(s), y)\right]^2 , \tag{4}$$

where $\mathcal{D} \subset \mathbb{N}^2$ is the image support. This expression is non-convex with respect to the displacement field u. Thus, to avoid a non-convex minimization, we consider a Taylor expansion of the non-linear term $I_r(x - \bar{u}, y)$ around an initial estimate $\bar{u}$ [11]. We note that the initial value $\bar{u}$ may be obtained from a correlation based method or from previous iteration within an iterative process.

This linearization leads to the following convex quadratic criterion:

$$J_{\mathcal{D}}(w) = \sum_{s \in \mathcal{D}} [L(s)\ w(s) - r(s)]^2 \tag{5}$$

where

$$\begin{aligned} w &= (u, v)^\top, \\ L_(s) &= [I_r^x(x - \bar{u}(s), y),\ I_l(s)], \\ r(s) &= I_r(x - \bar{u}(s), y) + \bar{u}(s)\ I_r^x(x - \bar{u}(s), y). \end{aligned}$$

$I_r^x(x - \bar{u}, y)$ is the horizontal gradient of the warped right image.

Optimizing this criterion is an ill-posed problem due to the fact that two variables have to be determined for each pixel and that the components of L may locally vanish. Thus, it is useful to incorporate additional constraints modelling available information to convert the problem to a well-posed one.

In this work, we address this problem from a set theoretic formulation [13], where each piece of information is represented by a convex set and the intersection of these sets, the feasibility set, constitutes the family of possible solutions. When the aim is to find an acceptable solution minimizing a certain cost function, the problem is formulated in a Hilbert image space $\mathcal{H}$ as follows:

$$\text{Find } w \in S = \bigcap_{i=1}^{m} S_i \text{ such that } J(w) = \inf J(S)\ , \tag{6}$$

where the objective $J : \mathcal{H} \to]-\infty, +\infty]$ is a convex function and $(S_i)_{1 \leq i \leq m}$ are closed convex sets of $\mathcal{H}$. Constraint sets can be modelled as level sets [14].

In what follows, we introduce our objective function as well as the considered convex constraints to solve the stereo matching problem within the set theoretic framework described above.

3.2 The Global Objective Function

According to the conditions of convergence of the algorithm we use [14] to solve (6), the objective function J must be strictly convex. However, since $J_{\mathcal{D}}$ in (5) is not strictly convex, we introduce an additional strictly convex term as follows:

$$J(w) = \sum_{\mathrm{s} \in \mathcal{D}} [L(\mathrm{s})\, w(\mathrm{s}) - r(\mathrm{s})]^2 + \alpha \sum_{\mathrm{s} \in \mathcal{D}} |w(\mathrm{s}) - \bar{w}(\mathrm{s})|_2^2\ , \tag{7}$$

where $\bar{w} = (\bar{u}, \bar{v})$ is an initial estimate, $|\cdot|_2$ denotes the Euclidean norm in $\mathbb{R}^2$ and $\alpha > 0$ weights the second term relatively to the first.

3.3 Convex Constraints

The construction of convex constraints is derived from the properties of the fields to be estimated. The constraint set arising from the knowledge of the disparity range values is $S_1 = [u_{\min}, u_{\max}]$. We note that in practice $u_{\min} \geq 0$ and $u_{\max}$ are often available. Additionally, in intelligent vehicle applications, the disparity map should be smooth in homogeneous areas while keeping sharp edges. This can be achieved with the help of a suitable regularization constraint. In this work, we investigate a regularization constraint based on the Total Variation which measures the amount of oscillations in the image u [15]. So, if a bound τ_u

is available on the Total Variation, controlling $\mathrm{TV}(u)$ restricts the solution to the following convex set:

$$S_2 = \{(u, v) \in \mathcal{H} \mid \mathrm{TV}(u) \leq \tau_u\} \,. \tag{8}$$

We have also considered prior knowledge for the multiplicative coefficient of the intensity change. In most scenes, brightness changes have slow variations in space. A Tikhonov-like quadratic term is appropriate to recover this kind of locally smooth field. Hence, we impose an upper bound on the quadratic norm of the gradient ∇v of the illumination coefficient v, so restricting the solution to the convex set

$$S_3 = \{(u, v) \in \mathcal{H} \mid \|\nabla v\|^2 \leq \tau_v\} \,, \tag{9}$$

where $\tau_v > 0$. In addition, it has been shown in [12], through experiments with real images, that v usually ranges between $v_{\min} = 0.8$ and $v_{\max} = 1.2$. The constraint set arising from this knowledge is $S_4 = [v_{\min}, v_{\max}]$.

In summary, we formulate the disparity estimation problem as the minimization of (7) on $S = \bigcap_{i=1}^{4} S_i$. Several methods have been proposed to solve this problem. In this work, we use the parallel block iterative algorithm developed in [14], which is particularly well adapted to our need. Indeed, this algorithm can be efficiently implemented on parallel computing architectures by adapting the number of constraint sets considered at each iteration to the number of parallel processors available. This enables to achieve real-time performance which is also mandatory for intelligent vehicle applications. Moreover, this algorithm offers great flexibility in the incorporation of several constraints. Details about this algorithm are given in [14].

4 Obstacle Detection Based on Disparity Image Analysis

The technique used in this paper for obstacle detection takes advantage of the reliability of the 3D information obtained from the preliminary disparity estimation stage described in the previous section. The result of such estimation on various stereo images acquired in different situations is shown in Figure 2.

Taking a look at the computed disparity maps made us consider a straightforward solution to detect obstacles. Indeed, we can observe from these images that the viewed landscape in a road environment consists of a plane surface with obstacles stated above the ground plane. We also observe that obstacles are characterized by vertical planes and that the ground is mostly horizontal. We note that vertical symmetry have already been chosen as a main distinctive feature to detect pedestrians [2].

In this work, we will use both observations made above in order to separate obstacle candidates from road surfaces. For this purpose, we first compute for each stereo point from the disparity image, a surface normal vector (n_X, n_Y, n_Z) in the vehicle coordinate system, which is depicted in figure 3.



Fig. 2. (Top) Left images. (Down) Computed disparity maps.

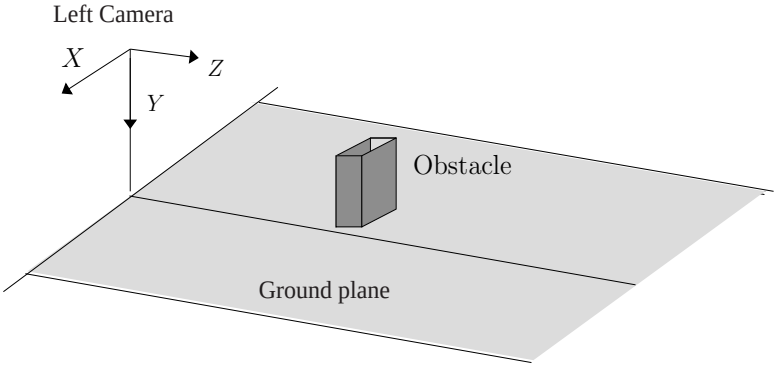


Fig. 3. The vehicle coordinate system



Fig. 4. Classification of stereo points from disparity images

Once surface normal vectors are computed, we separate obstacle points from road surface by considering that absolute values of n_Y close to 1 indicate a highly horizontal surface likely to be the ground and absolute values of n_Z close to 1 indicate vertical surfaces most likely to be obstacles. Figure 4 shows the result

of such classification superimposed onto the reference left image. Blue regions belong to the ground surface and red regions belong to the vertical surface planes which are likely to contain obstacles. These results demonstrate the effectiveness of the proposed method.

To effectively locate obstacles and evaluate their degree of danger, we finally compute the coordinates of pixels classified as belonging to vertical surfaces given the camera calibration parameters.

5 Conclusion

In this work, a stereo vision technique for obstacle detection has been proposed. Initially, we solve the stereo matching problem, formulated as a convex programming problem, through the minimization of a quadratic objective function over the intersection of multiple constraint sets. Then, we separate pixels belonging to the road surface from those belonging to obstacles based on the depth map. Using surface orientation as a classification criterion of 3D stereo data has proved to be a robust technique to detect any kind of obstacles at different situations.

References

1. Labayrade, R., Aubert, D., Tarel, J.P.: Real time obstacle detection in stereovision on non flat road geometry through v-disparity representation. In: IEEE Intelligent Vehicle Symposium, vol. 2, pp. 646–651 (2002)
2. Bertozzi, M., Broggi, A., Fascioli, A., Lombardi, P.: Vision-based pedestrian detection: will ants help? IEEE Intelligent Vehicle Symposium 1, 17–21 (2002)
3. Yu, Q., Arajo, H., Wang, H.: Stereo-vision based real time obstacle detection for urban environments. In: ICAR 2003. 11th International Conference on Advanced Robotics, Coimbra, Portugal, vol. 3, pp. 1671–1676 (2003)
4. Demonceaux, C., Kachi-Akkouche, D.: Robust obstacle detection with monocular vision based on motion analysis. In: IEEE Intelligent Vehicles Symposium, Parme, Italie (June 2004)
5. Toulminet, G., Mousset, S., Bensrhair, A.: Fast And Accurate Stereo Vision-Based Estimation Of 3D Position And Axial Motion Of Road Obstacles. International Journal of Image Graphics 4(1), 99–126 (2004)
6. Labayrade, R., Royere, C., Gruyer, D., Aubert, D.: Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner. Autonomous Robots 19(2), 117–140 (2005)
7. Giachetti, A., Campani, M., Torre, V.: The use of optical flow for road navigation. IEEE Transactions on Robotics and Automation 14(1) (1998)
8. Hu, Z., Lamosa, F., Uchimura, K.: A complete U-V-Disparity study for stereovision based 3D driving environment analysis. In: Fifth International Conference on 3D Digital Imaging and Modeling, Ontario, Canada, pp. 204–211 (June 2005)
9. Parent, M., De La Fortelle, A.: Cybercars: Past, present and future of the technology. In: ITS World Congress 2005, San Francisco, USA (2005)
10. Fusiello, A., Trucco, E., Verri, A.: A compact algorithm for rectification of stereo pairs. Machine Vision and Applications 12(1), 16–22 (2000)

11. Miled, W., Pesquet, J.C., Parent, M.: Disparity map estimation using a total variation bound. In: CRV 2006. Proc. of the 3rd Canadian Conference on Computer and Robot Vision, Qubec, Canada, pp. 48–55 (June 7-9, 2006)
12. Gennert, M.A.: Brightness-based stereo matching. In: Proc. IEEE 2nd Int. Conf. Computer Vision, pp. 139–143. IEEE Computer Society Press, Los Alamitos (1988)
13. Combettes, P.L.: The foundations of set theoretic estimation. Proceedings of the IEEE 81(2), 182–208 (1993)
14. Combettes, P.L.: A block iterative surrogate constraint splitting method for quadratic signal recovery. IEEE Trans. on Signal Proc. 51, 1771–1782 (2003)
15. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)

Cooperative Maneuver Study Between Autonomous Cars: Overtaking

Javier Alonso Ruiz[1], Javier Jiménez Gil[2], José Eugenio Naranjo[1], José Ignacio Suárez[2], and Blas Vinagre[2]

[1] Instituto de Automática Industrial, CSIC, Madrid, Spain
[2] Industrial Engineering School, University of Extremadura, Badajoz, Spain
{jalonso, jnaranjo}@iai.csic.es
javijgil@terra.es, {jmarcelo, bvinagre}@unex.es

Abstract. This research studies the overtaking maneuver in some representative situations. A simulator using Matlab Simulink connecting its outputs to a Virtual Reality module to show the complete overtaking performance from all points of view has been made. With this useful tool the parameters of the car controllers and decision making systems can be set up, and unforeseen new features of the overtaking maneuver can also be taken into account.

1 Introduction

Overtaking in two way roads is one of the most dangerous maneuvers. Making a mistake while doing this maneuver can lead to terrible accidents. That is why all the efforts to develop driving aids for this operation are one of the main issues of the Intelligent Transport System (from now ITS) area. The starting-point of our research in ITS was developing intelligent controllers for the automatic driving. Now, in order to achieve automated traffic systems in dedicated areas, we have aimed our research in the cooperative driving between several vehicles. The cooperative maneuvers already implemented have been the Adaptive Cruise Control, the Stop & Go driving and the Overtaking without an incoming flow of vehicles [3]. Once the possibility of finding an incoming car in the other lane is included, the risk of a dangerous situation makes the use of a simulator for the early steps of the research mandatory. A cheap and modular simulator has been made using Matlab toolboxes Simulink and Virtual Reality. To ensure a close collaboration between research groups, the decision and control modules are easily interchangeable and allow each group to test their own algorithms effortlessly. Moreover, the Virtual Reality output shows the overtaking evolution from all points of view. The simulator includes the kinematics of the car and a simplified model of its longitudinal and steering system dynamics. This work studies the overtaking maneuver in all possible situations. The goal is to improve the controllers and the decision making systems in real cars, and on top of that, to discover and take into account unforeseen new features of the overtaking maneuvers.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1151–1158, 2007.
© Springer-Verlag Berlin Heidelberg 2007

2 Simulator Overlook

The simulator is built on Matlab Simulink and includes the modeled cars with the kinematics and some dynamics. There is a module, the decision making subsystem, which opts for the best behavior in each moment. The fuzzy controller turns a desired behavior into the appropriate orders for the steering wheel, accelerator and brake. And the Virtual Reality output allows us to see the evolution of the maneuver from all points of view. The information used for the modeled cars in the simulator intended to be the same as that available from the real cars of our research. The vehicles have a differential GPS that provides a centimetrical precise position, so the coordinates of the simulated cars are without errors. The vehicles also have a radio-Ethernet connection to share its position, speed and intention with a high degree of reliability, so the simulated active car can decide upon the position, speed and intention of the rest of the cars. A detailed map of the driving zone with the speed limits and the forbidden overtaking zones is also integrated, and that information is simulated by setting the length limits to the endless straight lane road of the simulator. The map information and the data shared by the other vehicles permit a car to determine which vehicles are relevant to the maneuver in progress. In the simulator case all the vehicles are relevant and its information is always available.

3 Car Model

The modeled cars that appear on figure 1 are nonlinear single-track models, commonly called bicycle model [2], upgraded by steering wheel/wheel and

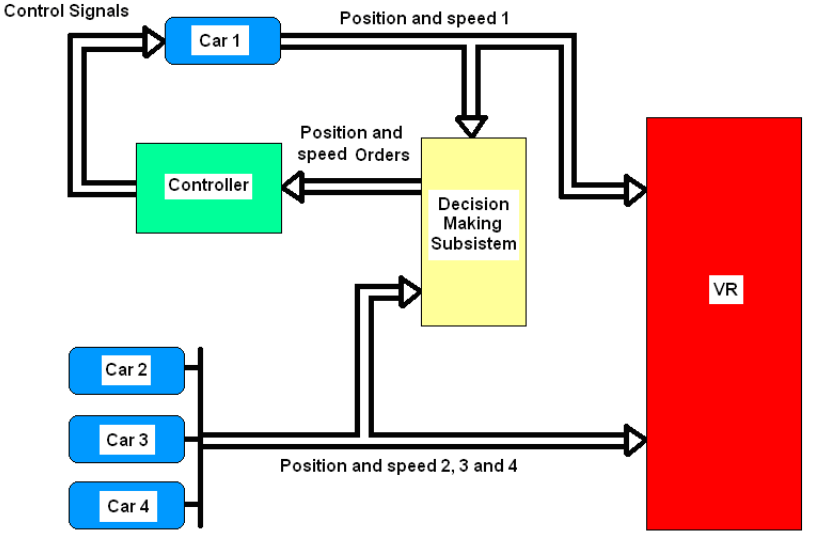


Fig. 1. Main diagram of the overtaking simulator

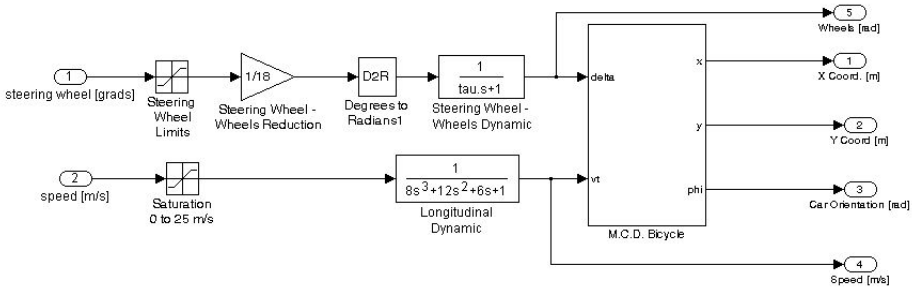


Fig. 2. Expanded view of the car model

longitudinal dynamics. The steering wheel/wheel dynamics avoid the instant change in the wheel angle. It is modeled by equation 1, obtained from experimental results with real cars in [4].

$$G(s) = \frac{1}{\tau s + 1} \tag{1}$$

The longitudinal dynamics avoid the instant change in the car speed. It is modeled by equation 2, obtained also in [4].

$$H(s) = \frac{1}{8s^3 + 12s^2 + 6s + 1} \tag{2}$$

4 Decision Making System

The essence of the decision making system can be explained with figure 3. This module is an automaton where each state is a desired car behavior. The transitions between states depend on processed information of the position, speed, acceleration of the cars involved in the overtaking maneuver. The active car (the overtaking car) position, speed and acceleration are used to compute the time needed to overtake the preceding cars (one or more, depends on the scenario). And the position, speed and acceleration of the incoming car in the left lane (the active car is in the right lane, as continental Europe drives on the right lane) are used to calculate the time to reach the active car. Using this measures the active car can, opt for overtaking or following the preceding car, or, once the overtaking has been started, for aborting the maneuver. The available modes for the active car are: Single navigation, Overtaking, Following the preceding car or Emergency overtaking abort. Each behavior has its own outputs like the reference line (imaginary line in the middle of the lane), the desired speed and the activation of the Adaptative Cruise (from now on, ACC) to follow the preceding car. These outputs set the goals for the fuzzy controller. In Single navigation mode, the active car remains in its lane and maintains its desired speed until it is near to the preceding car. In Overtaking mode, the active car establishes the

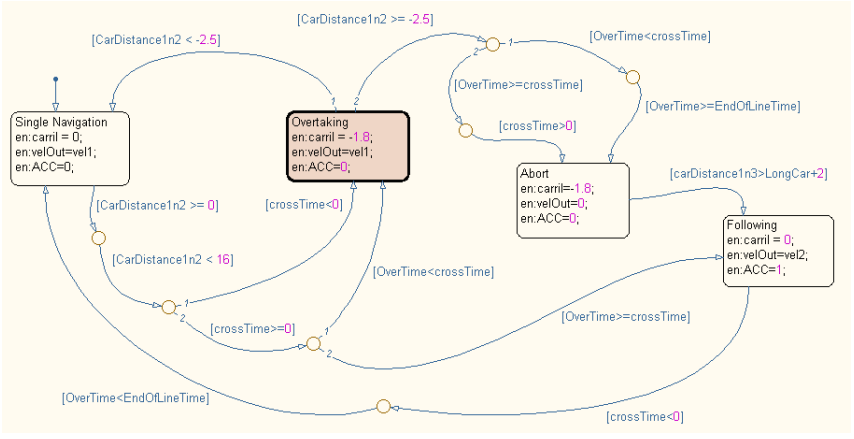


Fig. 3. Simplified Decision Making Flow Chart

left lane as the reference line and maintains the desired speed as reference speed. Once the preceding car is overtaken the system returns to the Single navigation mode and the right lane becomes the reference line. When the active car can not overtake and must remain in his lane, the system maintain the mode Following the preceding car. In this mode the car is referred to the right lane but the speed is controlled by the fuzzy controller to keep a safety time distance to the preceding car, being enabled the ACC behavior. If the incoming car speeds up when the active car is overtaking, the time measures change. In this case, aborting the maneuver and returning to a safety position are needed. When this happens, the system is set to Emergency overtaking abort mode. In this mode, the reference speed is set to zero to force a hard breaking. Once the active car is behind the car, that has been intended to be overtaken, the systems change to Following the preceding car mode. Its reference line is set to the right and speeds up to adjust its speed to the one of the preceding car.

5 Fuzzy Controller

The decision making system sets the reference line, the desired speed and enables or disables the ACC mode. Then, the fuzzy controller turns those goals into orders for the steering wheel, the accelerator, and the brake. This is done by minimizing the errors in the lateral displacement from the reference line, the orientation (angle), the speed, and the time gap with the preceding car. The fuzzy controller uses the Sugeno [5] method of fuzzy inference. MIN/MAX is used as inference method. In this step, the fuzzy input variables are used to calculate the truth value of the premises. For each rule, all the antecedent variables are computed to get the truth value of the premise, using "Min" as And method and

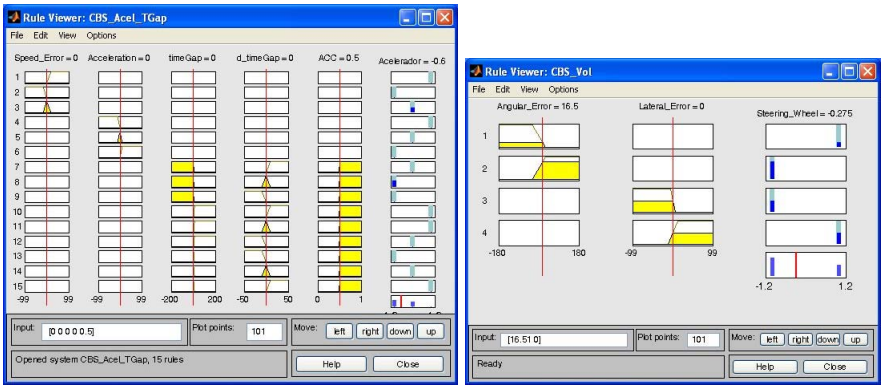


Fig. 4. Fuzzy controller for the accelerator and for the steering wheel

"Max" as Or method. Sugeno-style fuzzy inference uses Product as Implication method and Summation as Aggregation method. Once proper weighting has been assigned to each rule, the implication method is implemented. The consequent variables is reshaped using the product which scales the output fuzzy set. For each output variable the outputs of each rule are summed into a single fuzzy set as aggregation. For Defuzzification it is used the weighted average method to turn the output into a single number. The system state is described by means of this set of input variables:

1. Lateral error: the distance between the center of the back axis of the car and the line of reference (in meters).
2. Angular error: the difference between the car direction and the line of reference direction (in degrees).
3. Speed error: the difference between desired and real speed.
4. Acceleration: the speed variation over time.
5. Time Gap: the distance between the active car and the preceding car measured in seconds.
6. d_timeGap: the increase or decrease of the Time Gap.
7. ACC Signal: A signal that enables or disables the ACC rules.

The actuation over the system is defined by goals for the steering wheel, the accelerator, and the brake. The rules for the accelerator and the steering wheel are shown on figure 4.

6 Scenarios

To study the overtaking maneuver four scenarios were selected. The complexity of these scenarios increases from one to the following, reaching the final case which can be as complex as a daily situation.

6.1 Scenario 1: Overtaking in an Unlimited Straight Lane Road Without Incoming Vehicles

When the active car is near enough to the preceding car, it evaluates the risks of the overtaking maneuver. As there is no space or time limits, it decides to overtake. The decision module changes the reference line to the left lane and the fuzzy controller starts the maneuver. Once the overtaken car is some distance behind, the active car returns to the right lane. If the speed difference between the active car and the preceding card is big enough, the fuzzy controller only uses the steering wheel. However, if the speeds are similar, the fuzzy controller also uses the accelerator.

6.2 Scenario 2: Overtaking in an Unlimited Straight Lane Road With Incoming Vehicles

When the car is near enough to the preceding car, it evaluates the risks of the overtaking maneuver. As there is an incoming car, the decision subsystem must calculate if there is enough time to overtake. If there is time enough it overtakes like in scenario 1.If not, then it must follow the preceding vehicle until the incoming car passes. To follow the preceding car, the fuzzy controller must maintain a certain distance measured in time. In the simulation, the car changes its behav-

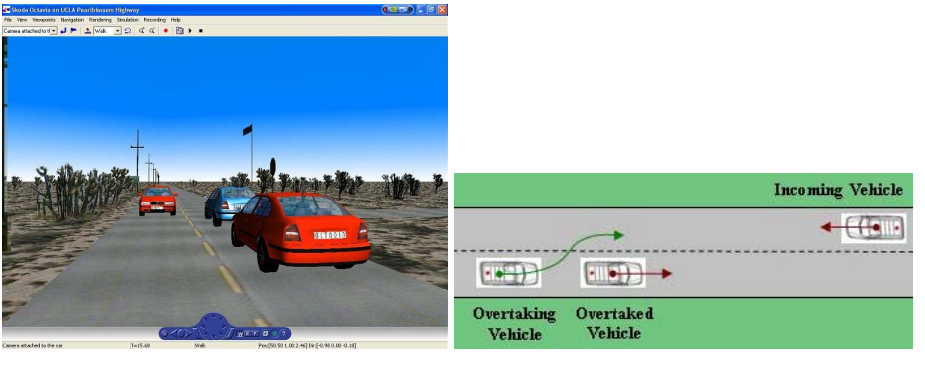


Fig. 5. Virtual Reality output for scenario 2, and scenario 2 diagram. Car models are from Ismar VR Company, and virtual environment is from UCLA.

ior from single navigation mode to Overtaking mode. When the incoming vehicle speeds up the evaluation of danger turns from a safe to an unsafe situation, and the car returns to the right lane and starts following the previous car. This emergency abort while overtaking, including the hard breaking, is a very interesting case of study. And knowing its limits and solutions is one of the main purposes of this work. A key issue of this simulation is the time margin left. This margin is recalculated constantly. If the incoming car speeds up, and the margin is not enough to finalize the overtaking maneuver, then the maneuver is aborted. Only

the simplest abortion maneuver is considered here, that is, hard breaking and returning to the right lane. There are other possibilities to consider when, time is running out for the active car, such as accelerating, especially if the overtaking maneuver is in very advanced stage; then much is still to be discussed.

6.3 Scenario 3: Overtaking with Space Limitations and Incoming Vehicles

As there are an incoming car, and a space limit (forbidden overtaking zones), the decision subsystem must calculate the minimum of the two time limits: the time to overtake before the incoming car passes and the time to reach the forbidden overtaking zone. Once this calculation has been completed, the maneuvers are like scenario 2.

6.4 Scenario 4: Overtaking with Space Limitations, Incoming Vehicles and Several Cars in the Same Driving Lane

The preceding cars in the same lane are considered a platoon. So the overtaking maneuver is considered for all of them. Not one by one. For the calculations of the time constrains the head and the tail of the platoon are considered. Once included this time constraint, the maneuvers are like scenario 3. In this scenario, the probability that time is run out before finalizing the overtaking maneuver is higher. Then, different options are open to discussion: accelerating to finalize, docking in the platoon (in which position?) and breaking to follow the platoon.

7 Conclusions

The simulator helps in tuning the real car controllers. The knowledge acquired by tuning the simulator can be very useful with real cars. Also helps choosing the decision making subsystem rules and constrains. The real car dimension becomes relevant. The car is not longer being represented like a point and a vector but it is seen and handled like a 3 dimensional object. It is also a useful tool to investigate the reduction of the time spent in the overtaking, the increase of the safety when overtaking, and the alternatives in an overtaking abort. Using the simulator, unexpected dangerous situations can be discovered while modifying the cars speeds and positions. Also relatively complex situations can be seen while playing with the simulation parameters.

References

1. Gven, B.A., Kural, E.: Adaptive Cruise Control Simulator. IEEE Control Systems Magazine 26(3), 42–55 (2006)
2. Gven, B.A., Gven, L.: Robust steer-by-wire control based on the model regulator. In: Proc. Joint IEEE Conf. Control Applications and IEEE Conf. Computer Aided Control Systems Design, Glasgow, pp. 435–440. IEEE Computer Society Press, Los Alamitos (2002)

3. Naranjo, J.E., Reviejo, J., Gonzlez, C., Garca, R., de Pedro, T.: Overtaking Maneuver Experiments with Autonomous Vehicles. In: Proceedings of ICAR 2003. The 11th International Conference on Advanced Robotics, Coimbra, Portugal (June 30 - July 3, 2003)
4. Surez, J.I., Vinagre, B.M., Chen, Y.Q.: A Fractional Adaptation Scheme for Lateral Control of an AGV. In: FDA 2006. Proceedings of the 2nd IFAC Workshop on Fractional Differentiation and its Applications, Porto, Portugal, July 19-21, pp. 212–217 (2006) ISBN: 972-8688-42-3 / 978-972-8688-42-4
5. Sugeno, M.: Industrial applications of fuzzy control. Elsevier Science Pub. Co., Amsterdam (1985)

Trajectory Planning in a Crossroads for a Fleet of Driverless Vehicles

Olivier Mehani[1] and Arnaud de La Fortelle[1,2]

[1] Inria Rocquencourt, Imara Team, Domaine de Voluceau, BP 105, 78153 Le Chesnay Cedex, France
{olivier.mehani,arnaud.delafortelle}@inria.fr
[2] École des Mines de Paris, CAOR Research Centre, 60 boulevard Saint-Michel, 75272 Paris Cedex 06, France

Abstract. In the context of Intelligent Transportation Systems based on driverless vehicles, one important issue is the passing of a crossroads. This paper presents a supervised reservation system. Extending previous works in this direction, the proposed algorithm determines trajectories and speeds for all the vehicles willing to pass the intersection. The work is separated into tasks shared between the vehicles and an infrastructure in charge of the crossroads. This paper describes the characteristics of the algorithm, as well as considerations which led to it. Simulation results are given and support the argument that this algorithm is probably suboptimal but still performs well. Finally, this work shows there is place for ameliorations and hints to further improve the algorithm are given.

Keywords: intelligent transportation system, cybercars, driverless vehicles, crossroads, time-space reservations, trajectory planning, speeds scheduling.

1 Introduction

In the context of ITS (Intelligent Transportation Systems) based on driverless road vehicles, one interesting and non-trivial problem is that of the passing of crossroads. At an intersection between two roads, the risk of collision is much higher due to the presence of other vehicles not going in the same direction, and potentially crossing each other's path.

It is necessary to find safe algorithms allowing every vehicle to pass this sensitive place without colliding. Moreover, it is important, while doing so, to keep efficiency in mind, for the vehicles not to wait forever before entering and passing the intersection. Every vehicle must pass the crossroads as fast as the safe possibilities allow.

An interesting approach to solve this issue is the use of reservations [2,3]. A supervising infrastructure is in charge of the intersection, and attributes space-time reservations of parts of the crossroads to the upcoming vehicles.

Basing our work on said previous litterature, we try to develop an algorithm usable for driverless cybercars which are parts of a complete ITS, and reasonably

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1159–1166, 2007.
© Springer-Verlag Berlin Heidelberg 2007

implementable in an actual system. In section 2, we formalize the problem and define the desired outputs of the algorithm. The proposed algorithm is then described in section 3. Section 4 describes the ad-hoc simulator developed to test this algorithm, as well smulation results. The last section sets way for future works in order to improve and implement this algorithm in real hardware.

2 Problem Statement

2.1 Layered Control

In a transportation system with fully autonomous vehicles (or cybercars), a lot of tasks have to be executed to answer the demand of a customer to travel between two points. The framework of this work is an approach consisting in decomposing the planning into three levels, each of which using only a relevant subset of the information, thus reducing the complexity:

the macroscopic level, *e.g.* a city or a region;
the mesoscopic level, *e.g.* a city quarter;
the microscopic level, *i.e.* the surroundings of the cybercar.

At the macroscopic level, a *path* is computed. A path, is defined as a succession of edges (road segments) in the graph description of the road network. This is the level of fleet and roads management with a time scale ranging from half an hour to several hours.

At the mesoscopic level, paths are transmitted by the upper level and turned into *trajectories*. A trajectory is a precise space-time curve that the cybercar has to follow. Typical precisions are 10 cm and 1/10 s. The goal of this level is to produce trajectories for all the cybercars circulating in a given area. These trajectories have to be safe, mainly collision-free, but also efficient in terms of throughput (most notably deadlock-free).

At the microscopic level, the cybercar's control moves along the trajectory and ensures that none of these collisions which couldn't have been foreseen at the higher levels occur.

In this paper, the focus is put on algorithms for the mesoscopic level. More precisely, the study is presented on a simple crossroads (X junction). It has, however, been done with a much more general setting in mind which the developed concepts are supposed to deal with.

One key element of this type of algorithms is the *respect of the controllability constraints of the vehicles* (its dynamics, *e.g.* attainable speeds), for the actual vehicles to be able to follow the generated trajectories. A second key element is the *management of the shared resources*, *i.e.* the intersection.

More formally put, the problem is to attribute a shared resource (the crossroads) to each of the vehicles for a given time period, while distributing this resource the most efficiently possible in order not to block the vehicles or even reduce the speeds too much.

2.2 Desired Output

Whatever the algorithm used to determine the scheduling, the expected output must be usable by a vehicle control algorithm, that is by a microscopic level. The data provided by the considered layer should, thus, be composed of position and speed (or time) information.

It is also necessary to have this information *in advance* so that the microscopic level can plan the accelerations or decelerations that may be needed.

3 Reservation Algorithm

3.1 Simple Crossroads Model

A crossroads is basically the intersection of two roads. Considering the most regular one, a single vehicle already has the option of choosing between three directions (Fig. 1, left). Doing so, the vehicle will have to occupy various contiguous time-space places of the crossroads. When more than one vehicle are willing to pass the crossroads at the same time, it is increasingly hard to determine a safe schedule ensuring that there will be only one vehicle at a given time-point, taking into account its physical dimensions.

3.2 Reservation of Parts of the Intersection

In [2], Dresner and Stone proposed to use square *patches* as the parts of the road space to reserve. They have, moreover, shown that the smaller the squares were, the more vehicles could pass the crossroads in a given time period. This is due to the fact that, with smaller patches, it is possible to reserve and free the needed space on the intersection more precisely, thus letting unneeded space available to other vehicles.

It is the authors' feeling, however, that this might not scale well to large crossroads or number of vehicles. Trying to get a better precision by reducing

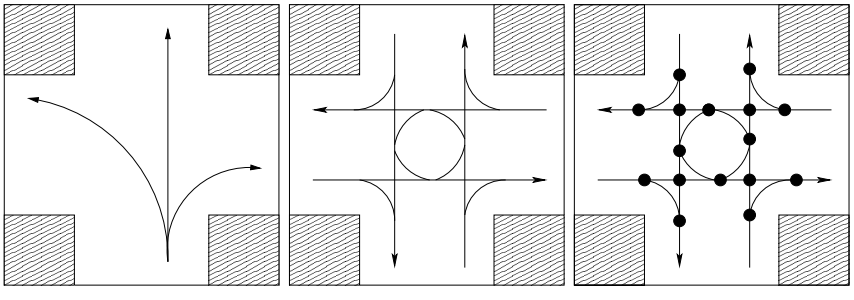


Fig. 1. A regular crossroads. Left: each vehicle entering it has three possibilities to get out. Center: trajectories that vehicles are able to, and *may*, follow. Right: only a small subset of critical points on the trajectories is where the risk of collision exists.

the reservable squares' size, thus increasing their number, would increase the charge on the supervisor, even if no collision could happen on some of the squares in normal operations (*e.g.* a part of the road where vehicle can only go in one direction). Reducing the set of reservable zones to those which actually need be reserved seems to be an appropriate way to increase the scalability of the system.

Given the dynamic constraints of the vehicles, it is possible to determine attainable 2-dimensional traces on the road (Fig. 1 on the previous page, center) that they are able to drive along to reach their target outgoing lane. One can see that only parts of the 2D traces are collision-prone. These are the places where a box (representing the vehicle) intersects another box placed anywhere on another trace. We further reduce the complexity by considering only the points where two or more traces intersect. This set is called the *critical points* (Fig. 1, left) and is proposed as the items to place reservations on in the intended algorithm. This simplifying assumption is justified by the simulations (using safety margins), where no collision occured. Of course a mathematical analysis should yields these safety margins from the geometry.

3.3 Outline of the Algorithm

Two types of entities are involved in the reservation process: (1) the vehicles, which place the reservations and adapt their speed in consequence, (2) the supervisor, which accepts or refuses a reservation request depending on the current schedule for the critical points. Each entity is supposed to be running a software agent able to execute the algorithm, which is, thus, separated into two distinct complementary parts.

Both intervenants retain useful information . The supervisor holds the geometry of the crossroads (*i.e.* the 2D traces) and maintains a list of all the current reservations on every critical point. The vehicle has information about its dimensions, and attainable speeds and accelerations. The agents need to share this information properly in order to successfully place a reservation.

The proposed method can be outlined as a three-steps algorithm (Fig. 2 on the facing page):

1. A vehicle arrives close to the crossroads and requests the crossroads geometry from the supervisor (*i.e.* the 2D traces and critical points);
2. According to its speed, it builds a reservation request which is sent back to the supervisor;
3. The supervisor decides whether the request is acceptable or not and informs the vehicle agent.

3.4 Building a Reservation Request

The following assumes a constant speed during one reservation *i.e.* , the vehicle is not accelerating, thus not integrating speed changes into the expected times computations. This assumption is made to simplify the problem solving, but removing it, in subsequent studies, may give more efficient results.

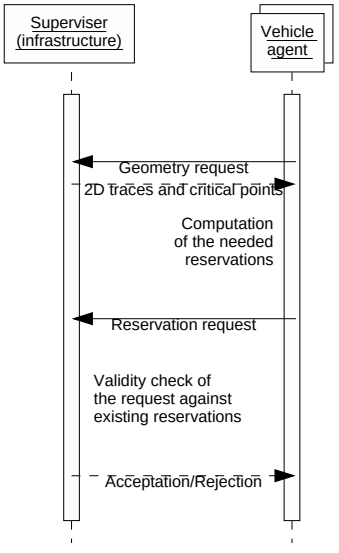


Fig. 2. A synopsis of the communications between a reservation agent and the crossroads supervisor. The proposed reservation scheme heavily relies on information exchanges between the supervisor and vehicles.

The vehicle arriving to an intersection has to compute the speed at which it will enter and pass it. To do so, it needs information about the crossroads' geometry. Moreover, it's the vehicle's role to try and place reservations on the critical points. The first step of the algorithm then consists of asking the infrastructure in charge of the crossroads to send these parameters.

Once all needed information is available, the *expected time of arrival* (ETA) to every critical point along the trajectory is computed. Knowing its current speed, the vehicle agent can also compute how long it will take to pass a critical point (*i.e.* the time period from when the front of the car starts being at the point to when its back enventually leaves it), the *expected time to pass* (ETP).

With both these values, a vehicle can determine the starting (1) and ending (2) times for each critical point it has to reserve.

$$t^{\text{start}} = ETA - \frac{sf\,ETP}{2}, \tag{1}$$

$$t^{\text{end}} = ETA + \frac{sf\,ETP}{2}. \tag{2}$$

The additional security factor $sf > 1$ is higly recommended in order to reserve a critical point a short time in advance, and free it a bit later than needed.

The reservation request is then built from these computations. It contains, for each critical point on the trace, the t^{start} and t^{end} parameters.

An improvement of this scheme, as derived from previous consideration about the construction of critical points, may be to reserve not only critical points on the trajectory, but also the *neighboring* ones. A vehicle is, indeed, not a punctual object and needs space around it to be able to do its maneuvers safely. A reasonable heuristic to determine whether a critical point can be considered as neighboring or not is to check its distance to the current on-trace critical point. If this distance is lower than the vehicle's half length, it can be considered as neighboring. In this situation, a reservation with the same timing values as the neighboring on-trace critical point should be requested at the same time.

3.5 Checking the Validity of a Request

The supervisor is in charge of accepting or rejecting a reservation. To do so, it keeps track of all the critical points of the crossroads and the times at which each is already reserved.

Upon an upcoming reservation request, it will sequentially check, for all of the critical points to be reserved, that the requested periods are free. In case *any* of them is not, the *whole* reservation is rejected. If no overlap is found, the reservation is accepted.

3.6 Behavior After a Request Has Been Answered

There are two outcomes to the reservation request: either it is accepted or refused. Depending on the supervisor's answer, the vehicle will behave differently:

the reservation is refused $\Rightarrow$ the vehicle slows down to stop *before* the first critical point while continuing to try and obtain a reservation;

the reservation is accepted $\Rightarrow$ the vehicle remains at a constant speed *or* tries to place new reservations assuming higher speeds in order to pass faster.

3.7 Comments About the Algorithm

It is important to note that, in the above algorithm, the work is clearly separated into three phases:

1. the vehicle agent's building of the reservation;
2. the supervisor's validation of the request;
3. communication between both entities.

This is very interesting as, as mentionned in [3], this means that the implementation of parts of this algorithm can be done in a fully separated way, as long as they respect the communication protocol. Taking the idea a bit further, this means that it is possible to implement, on either side, a completely different approach than those described above and, provided it is valid, still have a fully working system.

4 Simulation Results

In order to validate the algorithm and evaluate its efficiency, a simulator has been developed[1]. Written in Python, it simulates a crossroads and both vehicles and infrastructure cohabitating on it.

It has been developed as several modules, with the objective of making them easily interchangeable. The idea was to provide an algorithm-testing framework into which it would be easy to implement new crossroads scheduling policies to compare their efficiency. In order to test the performances of the proposed reservations algorithm, two other crossroads-passing policies have been implemented:

None which allows every vehicle to pass the intersection at full speed regardless of the collisions. It is supposed to be a good estimation of the lower bounds of the time needed to pass the crossroads. An algorithm getting time results close to those of this dummy policy could reasonably be considered a good one.

Polling treats the crossroads as a single atomic resource and only allows one vehicle at a time to pass it. This policy is 100% safe in terms of collision- and deadlock-freedom, but is intuitively not time-efficient.

Simulations of these two policies and the reservations based have been run for 100s[2] each. One can note (Table 1) that the dummy policy has a high throughput and relatively low passing times, while the polling policy only has one quarter of the throughput and much higher times, but no collision.

Table 1. The performances of the reservation algorithm compared to others. All results were obtained running the simulator for 100 (simulated) seconds with a 0.02s timestep.

	Time (s)			Collisions	Vehicles
	min	max	avg		
None	5.28	10.80	6.21	458	422
Polling	5.28	92.02	47.37	0	108
Reservations	5.28	16.82	9.79	0	134

The reservation algorithm, in comparison, behaves quite efficiently: no collisions have occured while the throughput was higher than that of the polling algorithm, and the times to pass were closer to those of the dummy policy.

This behavior is what was expected. The reservation algorithm detailed earlier was developed to provide a more efficient use of the crossroads space by distributing it between the vehicles in a much denser way than the polling algorithm does. Moreover, the fact that vehicles are *required* to have reserved a critical point before passing it is the key to avoid collisions on the managed space as long as the supervisor only accepts non-overlapping reservations.

[1] `https://gforge.inria.fr/frs/?group_id=424&release_id=951`

[2] In-simulation time, 0.02s timesteps

In this respect, one can consider that this reservation algorithm, as simulated here, is quite efficient and may be interesting to study and implement further.

5 Conclusion and Future Works

After simulating the proposed algorithm, it turns out that it is, indeed, an interesting scheduling method to improve the crossroads management in the context of fully automated cybercars. It has been shown that better results were obtained than with polling methods similar to traffic lights, as the intersection can be used by several vehicles at the same time thanks to the separation of the main resource into critical points.

The preliminary simulations which have been run, however, were assuming a lot of simplifying conditions which should be, in future works on this topic, removed in order to obtain a much more realistic simulation and implement the algorithm in actual cybercars.

A good improvement to consider is to remove the "constant speed" assumption and compute reservations for an accelerating (or decelerating) trajectory instead of constantly trying to place reservation at increasing speeds once the first one has been granted. Other ideas would be to have the vehicles generate more than one reservation request at a time (vehicle agent modification only), or let the supervisor propose (for resubmission) an acceptable reservation in case it rejected that of a vehicle

One important thing to do would also be to formally prove the deadlock-freedom of the algorithm, in the case of a perfect microscopic level, and provide ways to avoid those which may arise in a more realistic case (*e.g.* if a cat crosses in front of a car, preventing it from being able to continue *on time*, on its trajectory).

Aknowledgments. This work has been partially funded by the EC project CyberCars-2 [1].

References

1. European project CyberCars-2, `http://www-c.inria.fr/cybercars2/`
2. Dresner, K., Stone, P.: Multiagent traffic management: A reservation-based intersection control mechanism. In: The Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York, USA, pp. 530–537 (July 2004)
3. Dresner, K., Stone, P.: Multiagent traffic management: An improved intersection control mechanism. In: The Fourth International Joint Conference on Autonomous Agents and Multiagent Systems, Utrecht, The Netherlands, pp. 471–477 (July 2005)

Secure and Scalable Communication in Vehicle Ad Hoc Networks

Jan Nikodem and Maciej Nikodem

Wrocław University of Technology,
The Institute of Computer Engineering, Control and Robotics
Wybrzeże Wyspiańskieg 27, 50-370 Wrocław, Poland
{jan.nikodem, maciej.nikodem}@pwr.wroc.pl

Abstract. In this paper the Vehicle Ad Hoc Network (VANET) is examined. As widely agreed VANETs must rely heavily on node-to-node communication while ease of access has to be assured at the same time. Beside the performance ensuring data authentication is another concern in VANETs.

In our paper we focus on security aspects for the VANETs that aim to ensure data authentication and allow to ensure data secrecy. Ensuring data secrecy is not a standard feature of the VANETs since usually it is assumed that nodes in the network has to be able to receive all information send over the channel. However we think that it is necessary to ensure security against data eavesdrop and message forgery in some applications (e.g. police pursuits or military convoys).

Keywords: ad hoc networks, security, key distribution, node exclusion.

1 Introduction

Wireless ad hoc networks are perfect candidates for operating in environments without any network infrastructure, with difficult access or nodes in constant move (e.g. in Vehicle Ad Hoc Networks - VANETs). Comparing to other applications of the ad hoc networks (e.g. Wireless Sensor Networks WSNs) VANETs are not nearly as constrained as WSNs in terms of available energy, storage and computational capacity. On the other hand VANETs are characterized by high nodes mobility and rapidly changing topology of the network.

Above mentioned properties of the VANETs cause additional difficulties in clustering procedure and message routing [13]. Solutions to these problems have been presented for VANETs with no access restrictions, i.e. where every node can join the network at any time. Some restrictions have been introduced by Little and Agarwal [11] who propose to distinguish traffic direction and associate vehicles with their direction.

In our paper we focus on VANETs with limited access that can be used in police pursuits, military convoys or similar groups of vehicles where message authentication and secrecy have to be ensured. This obviously can be achieved using encryption but standard encryption algorithms cannot be used in straight

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1167–1174, 2007.
© Springer-Verlag Berlin Heidelberg 2007

forward way. This is because node's high mobility and changing topology of the network requires procedures for network management (i.e. how to add new and exclude broken or compromised nodes). This yields another problem of exchanging and distributing the encryption keys between nodes of the network in secure, effective and a very fast way.

2 Related Work

The challenge in this paper is a key distribution i.e. how to send keys to each node of the network efficiently (with small communication and little computation and storage overhead) and securely over the broadcast channel.

All existing methods for group rekeying schemes are based on one of four approaches (e.g. [10,12]): certificate based approach, Generalized Diffie-Hellman (GDH) based approach, broadcast exclusion approach and probabilistic approach. Unfortunately all of them have drawbacks.

Public key approaches (certificate and GDH based) have large computational complexity and are not efficient in case of large networks [1,15]. Another disadvantage of these methods is a large communication overhead when nodes are excluded.

Broadcast exclusion protocols [5,7,9,16] typically assume that the network structure is known which is hardly possible to fulfil in VANET's where nodes are in constant move. Another disadvantage of the broadcast exclusion protocols is that they were proposed to solve problems in one-to-many communication (e.g. pay-TV) with only one broadcaster. In case of the ad hoc network one central node is likely to become a bottleneck of the whole network. The advantage of broadcast communication is a low communication and storage overhead. Computational complexity is usually larger than in case of other approaches but it can be reduced using different techniques (e.g. pre-processing [16]).

Probabilistic approach is based on large set of symmetric keys [18]. Each node in the network stores random subset of these keys. Assuming that each node obtains keys independently at random, knowing the number of keys in the set and number of nodes we can estimate the probability that after deployment neighbour nodes will share at last one common key. This happens with high probability (usually about 95–99%) allowing neighbour nodes to communicate directly and securely. If direct communication is impossible then nodes establish a multi-hop communication path using nodes that share at last one common key. In these schemes both storage and communication complexity is higher then in case of the broadcast exclusion while computation complexity is kept small.

Another solution is a key pre-distribution algorithm presented by Blom [2] that utilises linear codes over $GF(q)$. Two basic concepts of this scheme are: a generator matrix G known to all nodes; a secret, random, and symmetric matrix D. The secret data stored in the device during pre-distribution consists of one row of the $(DG)^T$ matrix. Pairwise keys, K_{ij}, are defined as elements of a matrix $K = (DG)^T G$, and K is symmetric thus $K_{ij} = K_{ji}$. Using secret data and matrix G, node i can compute one column of the K – say the i-th

column, and since K is symmetric node i knows also i-th row of the K. Also node j, using his secret data, can compute one column and one row of the K – say the j-th column and row. These columns and rows have exactly two common elements – K_{ij} and K_{ji}, which are equal. In this way, both nodes, knowing only their identities, the matrix G and secret data, can compute a secret key for communication.

Polynomial-based key pre-distribution was first proposed by Blundo et al.[3] who showed that a scheme based on symmetric polynomials in k variables of degree t is t-resilient and optimal. Their work was further extended by Liu and Ning [12] and Sadi et al.[14] who brought together polynomial-based and random key pre-distribution schemes – so called polynomial pool-based key pre-distribution.

In our paper we describe a group rekeying scheme that takes advantage of solutions presented so far. We propose to use key pre-distribution and broadcast exclusion protocol, which allow to manage network members (adding and eliminating nodes), ensure secure communication between nodes and allow to establish encryption keys efficiently. Our proposal, together with other works on routing [4,8] and network organization, aims to build the VANET that organises itself and acts as one structure.

3 Network Assumptions

Throughout rest of the paper we will assume that VANET consist of a group of nodes joining and leaving the network during its lifetime. Beside legal nodes which belong to the network and are entitled to communicate with each other securely, there are also illegal nodes. Illegal nodes are nodes that are not entitled to communicate with the network. Among illegal nodes there are nodes that belong to other network and malicious nodes that aim to compromise VANET's security.

In our paper we assume that security is compromised if illegal node (or nodes) successfully join the network, eavesdrop messages, send faked messages (i.e. are capable impersonate itself) or manage to drop or jam selected messages send between legal nodes of the network.

Concerning legal nodes we assume that nodes are capable of computing modulo exponentiation and store few kilobytes of data required to carry on cryptographic procedures.

4 Polynomial-Based Key Pre-distribution

In this section, we only deal with the basic scheme of key pre-distribution and consider how to establish pairwise keys between any two neighbour nodes. There are other types of keys, such as individual keys, group keys or routing keys. However, all of these keys can be established in many different ways when pairwise keys are known (e.g. [6,17]).

From now on, all computations are carried in Galois field $GF(q)$ where q is a prime integer. For simplicity, we assume that q and the output of the hash function $HASH$, used in our protocol, have the same bit length (e.g. 128 bits).

4.1 Pre-processing

Nodes of VANET are initialised before they are put into the vehicles. The initialisation consist of two steps. First, a random master key K_m is selected and surface $f(x, y)$ is generated. Afterwards each node i gets two polynomials

$$\begin{aligned} g_i(x) &= f(x, i) \\ h_i(y) &= f(i, y) \end{aligned} \tag{1}$$

of degree t. Both polynomials and the master key are stored in node's memory. According to (1) polynomials $g_i(x)$ and $h_i(y)$ are generated as an intersection of the surface $f(x, y)$ with planes $y = i$ and $x = i$ respectively. Moreover, if we ensure that coefficients of these polynomials are equal then only $O(t)$ memory is required to store them.

To achieve above properties of $g_i(x)$ and $h_i(y)$, it is necessary to generate the surface $f(x, y)$ according to the following formula:

$$f(x, y) = \sum_{i=0}^{t} a_i(y) x^i, \tag{2}$$

where coefficients $a_i(y)$ are polynomials of degree t in variable y. Therefore the resulting surface equals:

$$f(x, y) = \sum_{i=0}^{t} \sum_{j=0}^{t} a_{ij} y^j x^i = \sum_{j=0}^{t} \left(\sum_{i=0}^{t} a_{ij} y^j x^i \right), \tag{3}$$

and can be equivalently represent in form of matrix multiplication:

$$f(x, y) = Y^T \cdot A \cdot X = \begin{bmatrix} 1 \\ y \\ y^2 \\ \vdots \\ y^t \end{bmatrix}^T \begin{bmatrix} a_{00} & a_{10} & a_{20} & \dots & a_{t0} \\ a_{01} & a_{11} & a_{21} & \dots & a_{t1} \\ a_{02} & a_{12} & a_{22} & \dots & a_{t2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{0t} & a_{1t} & a_{2t} & \dots & a_{tt} \end{bmatrix} \begin{bmatrix} 1 \\ x \\ x^2 \\ \vdots \\ x^t \end{bmatrix}$$

For constant $y = i$ the sum of elements from within k-th column of $Y^T A$ equals to the k-th coefficient of polynomial $g_i(x)$. Similarly for constant $x = i$ the $k-th$ element of AX equals to the k-th coefficient of polynomial $h_i(y)$. Precisely k-th coefficient of these polynomials equal:

$$g_{ik} = \sum_{j=0}^{t} i^j a_{kj}, \qquad h_{ik} = \sum_{j=0}^{t} i^j a_{jk}. \tag{4}$$

Finally, both polynomials $g_i(x)$ and $h_i(y)$ are equal if they are of the same degree and have corresponding coefficients equal. According to (4) this yields $a_{jk} = a_{kj}$. Therefore if we take symmetric matrix of coefficients A and generate surface according to (3) then polynomials (1) have exactly the same coefficients. This also means that $O(t)$ memory is large enough to store polynomials (1).

5 Securing Network Communication

5.1 Pairwise Key Determination

Since every two nodes i, j from the VANET poses polynomials (1) that belong to the same surface $f(x, y)$ thus they share information about two points in which polynomials intersect. This points can be computed independently by each node if identity of other one is known

$$g_i(j) = f(j, i) = h_j(i) \qquad h_i(j) = f(i, j) = g_j(i). \tag{5}$$

Note that two nodes share exactly one information since $f(i, j) = f(j, i)$ due to symmetrical matrix A used for key pre-distribution.

Polynomial-based key pre-distribution enables any two nodes knowing each other identity to get encryption key immediately without sending any part of the key between them. This ensures high security of key establishment.

5.2 Group Key Determination

When there is a group of nodes in the range they can use their polynomials to determine pairwise keys. Later on they can use this keys in order to establish one key for the whole group securely. However this is one of possible solutions, it yields large communication overhead and may cause additional concerns.

Contrary we propose to take advantages of fact that we may select one node from the group to be a group leader. Then all other nodes share exactly one and unique secret information with the leader and these information belong to the leader's polynomials. In other words leader knows all the secret information it shares with nodes from the group while each node knows only the secret information he shares with the leader (has no information about other secrets). Some readers may realise that in this way we get a secret sharing scheme. The leader knows a polynomial (which is the secret in the scheme) and each node knows one share of this secret.

If so then we can use protocols that are based on secret sharing. In particular we propose to use broadcast encryption protocols that enable to exchange secret key between the leader and group of share holders effectively and securely while managing the network membership at the same time.

Simple Broadcast Exclusion Protocol for WANETs. Taking advantages of fact that all nodes u_i from the group Ω poses exactly one and unique share of the leader's secret we can establish group key effectively and securely. This can be done as follows:

- leader u_l selects a prime p and a generator α of the multiplicative group Z_p,
- u_l selects random integers $r, q < p-1$,
- u_l determines a set $\Phi = \left\{(x, g^{r\alpha_{u_l}(x)})\right\}$ such that $|\Phi| = t$ and $x \notin \Omega$,
- leader selects random group key $K^G < p$ and encrypts it as $E(K^G) = K^G \cdot \alpha^{rg_{u_l}(q)}$,
- leader constructs and broadcasts a header:

$$H = \left\langle q, \alpha^r, E(K^G), \Phi \right\rangle$$

- upon receiving the header each neighbour node $u_i \in \Omega$ performs following operations:
  - computes $(\alpha^r)^{h_i(u_l)}$ (note that $\alpha^{rh_{u_i}(u_l)} = \alpha^{rg_{u_l}(u_i)}$),
  - adds a pair $\left\langle u_i, \alpha^{rh_{u_i}(u_l)} \right\rangle$ to the set Φ (note that from now on $|\Phi| = t+1$),
  - computes a Lagrangian interpolation in the exponent

$$D = \prod_{i\in\Phi} \alpha^{rg_{u_l}(u_i) \sum\limits_{j\in\Phi, j\neq i} \frac{q-u_j}{u_i-u_j}} = \alpha^{r \sum\limits_{i\in\Phi} g_{u_l}(u_i) \sum\limits_{j\in\Phi, j\neq i} \frac{q-u_j}{u_i-u_j}}$$

    where $LI(q) = \sum_{i\in\Phi} g_{u_l}(u_i) \sum_{j\in\Phi, j\neq i} \frac{q-u_j}{u_i-u_j}$ is a Lagrangian interpolation for polynomial $g_{u_l}(x)$ of degree t for $x = q$. Therefore $LI(q) = g_{u_l}(q)$.
  - computes the group key

$$K^G = \frac{E(K^G)}{D} = \frac{K^G \cdot \alpha^{rg_{u_l}(q)}}{\alpha^{rLI(q)}} = \frac{K^G \cdot \alpha^{rg_{u_l}(q)}}{\alpha^{rg_{u_l}(q)}}.$$

Using above procedure group leader u_l can send a group key to all nodes from the group Ω efficiently and securely. Moreover proposed protocol will work in exactly the same manner for any group of nodes (independently of number of nodes in the group).

Security of the protocol arises from the use of secret sharing scheme and broadcast exclusion protocol. Observe that leader sends only t points (elements of the set Φ) that convey information about his polynomials. This is not enough for any attacker to reconstruct these polynomials since at least $t+1$ points are required to reconstruct polynomial of degree t. Additionally points send over the channel are masked with random integer r and hidden in the exponent modulo prime p. Complexity of computing discreet logarithms for large p ensures that it is infeasible to determine points that belong to the leader's polynomials. On the other hand random integer (also unknown to anyone except the leader) ensures that it is impossible to use headers from previous transmissions to reconstruct future group keys. This allows to repeat proposed protocol infinitely many times and also enables the leader do manage the network membership.

When determining the set Φ leader can select some of its elements in such way that $x = u_i$ for some $u_i \in \Omega$. Remaining parts of the protocol are performed in exactly the same way as before. However, nodes which identifiers were used to create the set Φ are called excluded nodes and are not able to reconstruct the group key. In fact, after node u_i receives the header it adds its own

pair $\left\langle u_i, \alpha^{r h_{u_i}(u_l)} \right\rangle$ into the set Φ. In this way $|\Phi| = t + 1$ but there are only t unique pairs in the set since node's pair was put into the Φ by a leader. Therefore excluded node poses information that is insufficient to perform Lagrangian interpolation and thus compute the group key.

Broadcast exclusion is an effective and secure method of distributing the key to group of nodes. Additionally it enables the leader to manage the network membership by deciding which nodes are not entitled to reconstruct the group key. Nodes exclusion is limited by the degree of polynomials $\deg(g_i(x)) = \deg(h_i(y)) = t$. Therefore up to t nodes from the group can be excluded simultaneously. Another advantage of the scheme is its communication overhead that is independent of the number of nodes in the group.

Described scheme also ensures t-resilience which means that up to t nodes can be captured and reverse engineered without affecting the security of the whole network.

6 Conclusions

This paper presents a method to establish a secure communication in VANETs with polynomial-based key pre-distribution. Such possibility may be useful in many different situations when group of vehicles has to communicate in private (e.g. during police pursuits). Our proposal ensures that any two vehicles are able to determine unique pairwise key effectively knowing only each other identities. Moreover group of such vehicles can securely establish a group key using secret sharing and broadcast encryption schemes.

The proposed protocol also ensures low communication overhead by performing broadcast communication rather than one-to-one communication. This reduces the number of messages sent over the broadcast channel and reduces the probability of collisions and retransmissions. Moreover the communication overhead does not depend on the number of nodes in the group but on the parameter t.

References

1. Anton, E., Duarte, O.: Group key establishment in wireless ad hoc networks. In: Workshop on Quality of Service and Mobility (WQoSM) (2002)
2. Blom, R.: An Optimal Class of Symetric Key Generation Systems. In: Beth, T., Cot, N., Ingemarsson, I. (eds.) Advances in Cryptology. LNCS, vol. 209, pp. 335–338. Springer, Heidelberg (1985)
3. Blundo, C., De Santis, A., Herzberg, A., Kutten, S., Vaccaro, U., Yung, M.: Perfectly Secure Key Distribution for Dynamic Conferences. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 471–486. Springer, Heidelberg (1993)
4. Buchegger, S., Le Boudec, J.-Y.: Cooperative Routing in Mobile Ad Hoc Networks: Current Efforts Against Malice and Selfishness, Mobile Internet Workshop. In: Informatik 2002. LNCS, Springer, Heidelberg (2002)

5. Castelluccia, C., Saxena, N., Yi, J.H.: Self-configurable Key Pre-distribution in Mobile Ad Hoc Networks. In: Boutaba, R., Almeroth, K.C., Puigjaner, R., Shen, S., Black, J.P. (eds.) NETWORKING 2005. LNCS, vol. 3462, pp. 1083–1095. Springer, Heidelberg (2005)
6. Dimitriou, T.D.: Securing Communication Trees in Sensor Networks. In: Nikoletseas, S.E., Rolim, J.D.P. (eds.) ALGOSENSORS 2006. LNCS, vol. 4240, pp. 47–58. Springer, Heidelberg (2006)
7. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
8. Filipovic, A., Datta, A., McDonald, C.: Low-Latency Routing in Ad Hoc Wireless Networks Using Finite Horizons. In: Chen, G., Pan, Y., Guo, M., Lu, J. (eds.) Parallel and Distributed Processing and Applications - ISPA 2005 Workshops. LNCS, vol. 3759, pp. 422–433. Springer, Heidelberg (2005)
9. Kim, C.H., Hwang, Y.H., Lee, P.J.: Practical Pay-TV Scheme using Traitor Tracing Scheme for Multiple Channels. In: Lim, C.H., Yung, M. (eds.) WISA 2004. LNCS, vol. 3325, pp. 265–279. Springer, Heidelberg (2005)
10. Kui, R., Gang Yao, Y.: Key Agreements in Ad Hoc Networks. In: Chinacrypt 2004. Proc. of Advance in Cryptology, Shanghai (2004)
11. Little, T.D.C., Agarwal, A.: An information propagation scheme for VANETs. In: Proceedings of 2005 IEEE Intelligent Transportation Systems, pp. 155–160. IEEE Computer Society Press, Los Alamitos (2005)
12. Liu, D., Ning, P., Sun, K.: Efficient Self-Healing Group Key Distribution with Revocation Capability. In: CCS 2003, ACM Press, New York (2003)
13. Mabiala, M., Busson, A., Veque, V.: Inside VANET: Hybrid Network Dimensioning and Routing Protocol Comparison. In: VTC 2007-Spring. 65th IEEE Vehicular Technology Conference, Dublin, Ireland, pp. 227–232 (2007)
14. Sadi, M.G., Kim, D.S., Park, J.S.: GBR: Grid Based Random Key Predistribution for Wireless Sensor Network. In: ICPADS 2005, pp. 310–315. IEEE Computer Society, Washington, DC, USA (2005)
15. Steiner, M., Tsudik, G., Waidner, M.: Diffie-Hellman Key Distribution Extended to Group Communication. In: ACM Conference on Computer and Communications Security, pp. 31–37 (1996)
16. Watanabe, Y., Numao, M.: Multi-round Secure Light-Weight Broadcast Exclusion Protocol with Pre-processing. In: Snekkenes, E., Gollmann, D. (eds.) ESORICS 2003. LNCS, vol. 2808, pp. 85–99. Springer, Heidelberg (2003)
17. Zhu, S., Setia, S., Jajodia, S.: LEAP: Efficient Security Mechanism for Large-Scale Distributed Sensor Networks. In: CCS 2003, ACM Press, New York (2003)
18. Zhu, S., Setia, S., Xu, S., Jajodia, S.: GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad Hoc Networks, Technical report ISE-TR-02-01 (2004)

Speed Supervisor for Intelligent Vehicles

J.P. Carrasco, A. de la Escalera, and J.Mª. Armingol

Intelligent Systems Lab. Universidad Carlos III de Madrid. Leganés, Madrid, 28911. Spain
{juanpa, escalera, armingol}@ing.uc3m.es

Abstract. Advance Driver Assistance Systems help the driver with functions as communications, road mark detection, road sign recognition or pedestrian detection. A real time system that warns the driver in case of anomalous speed is described in this article. The speed of the vehicle is obtained from a GPS and at the same time a Computer Vision System is used to recognize danger, prohibition, yield and end of limit road signs. If the speed is not adequate to the recognized road sign the driver will be warned by a message from the vehicle's loudspeakers.

Keywords: Intelligent Vehicles, Computer Vision, Road Sign Recognition.

1 Introduction

As the number of vehicles on the roads has risen, there has been an increase in the number of traffic accidents. In Europe, 127.000 people are killed and at least 2.4 million are injured or disable [1]. Moreover, the associated cost is about 2% of the Gross Domestic Product of the industrialized countries. Many efforts are being made in different ways to deal with this current situation: improvement of the roads, conscience campaigns and active and passive security systems. The Implementation of different Advance Driver Assistance Systems is found to be a good way to deal with dangers and accidents and to inform giving advice to the driver when necessary. In this way, a new GPS-Road Sign Recognition System (RSR) is presented (fig.1). The vehicle's speed is obtained by a GPS and used as one of the inputs of the system; at the same time a Computer Vision System recognizes the danger, prohibition, yield and end of limit road signs. If the speed is inappropriate for the recognized road sign, a message is sent via the vehicle's loudspeakers to warn the driver. The messages will be emitted for the following situations: the vehicles speed is over the limit defined by the road sign, the speed of the vehicle is below the minimum speed allowed, the speed of the vehicle is over an input threshold set by the user/driver in case of danger or yield road sign. The system is onboard the experimental platform IVVI [2] (fig.10), this allows system testing in real situations. Since the system works in real time, the driver is continually updated with important information.

Few examples of the use of GPS and RSR System are found in the literature. A PDA and a digital camera are used to recognize the road signs creating a database with the sign and the position given by a GPS [3], however, the camera must be pointed directly at the road signs, off-site post-processing is needed and does not

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1175 – 1182, 2007.
© Springer-Verlag Berlin Heidelberg 2007



Fig. 1. GPS-RSR System. The system compares speed and road sign and warns the driver when necessary.

work in real time. Another system uses a GPS to generate a database of images and positions of objects of interest identified from video [4] also this system does not work in real time.

RSR Systems usually divide the tasks into two stages: detection and recognition of the road sign. The first one, commonly uses different colour spaces in order to enhance road signs typical colours. RGB [5] is easily implemented but usually needs post-processing as histogram enlargement to minimize lighting changes, or morphological transformation as opening or closing to reduce noise. HSI, HSL, HSV [6], [7], [8], are very similar and commonly used since the Hue component does not vary too much under distance or illumination changes. Some authors do not find colour segmentation reliable so prefer using Greyscale since it is invariant to the illumination changes and shadows [9]. Achromatic Spaces [10] have been defined in order to separate colour and non colour regions.

In the recognition stage, neural networks [11] are used because of their flexibility since they can deal with occlusions, rotations or road sign aging; but they have the disadvantage of a high computational cost. Normalized correlation [6], [12] allows fast algorithms implementation, obtaining high scores in detection and recognition. The drawback is that depends too much in the models used and on image quality.

2 The System

The PDA is continuously receiving information from a GPS by bluetooth antenna (fig. 2). These data follow the protocol of the National Marine Electronics Association (NMEA) [13](fig. 3), consisting in different ASCII coded messages which contains the whole information related to the global positioning system. In the PDA the messages are filtered, since only the messages Global Positioning fix data (GGA) and Course over Ground and Ground Speed (VTG) are used, and separated to form a new data string which will contain the data required for the Vision System. Since the GPS sends information about data reliability, as the Horizontal Dilution of Precision or number of satellites emitting, it can be used to reject erroneous data.

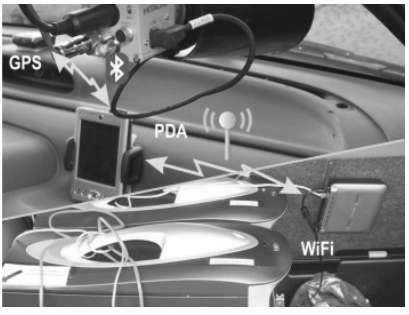


Fig. 2. The communication system. The PDA receives data from a GPS and it broadcasts it to the PC's.

$GPGGA,**161229.487**,3723.2475,N,12158.3416,W,**1**,**07**,**1.0**,9.0,M, , , ,0000*18

$GPVTG,**309.62**,T, ,M,0.13,N,**0.2**,K*6E

Fig. 3. Examples of GPS messages. Data in bold are part of the final data string: time, position fix indicator, satellites used, Horizontal Dilution of Precision, course and speed.

The new string will be sent to the PC's, where the image analysis is performed, by wireless connection.

In the Vision Application (fig.4) a camera is in continuing RGB image acquisition (A). Two different colour spaces will be used to deal with danger, prohibition and end of limit road signs. In the first two cases, HSI enhancement of the red color is implemented, a grayscale image is obtained where pure red is given the value 255 (B). HSI gives better results since its components do not vary significantly under distance or light changes. In the case of end of limit signs an achromatic space defined from the RGB cube is used. After enhancement, normalized correlation is used, for which purpose, models are needed to represent the road sign borders (C); taking into account the effects of distance, weather or camera movements, different blurred border models that represent reality better have been created (D). Applying Normalized Correlation similarity scores can be obtained. If they are over a given threshold it means that a possible road sign target (E) has been found. The RGB image is cut under the maximum model score (F) and normalized (G). Normalized correlation is applied again over a template containing the whole road signs plates (H). The output is given by the image of the road, the recognized road signs and the messages to the driver if necessary (I).

Finally, a couple of data is obtained: [Speed, Recognized Road Sign]. If the speed is not adequate for the prohibition or end of limit road sign, or the user threshold set for the danger or yield road signs, the system will write a message on the screen and acoustically warn the driver by the vehicle's loudspeakers.

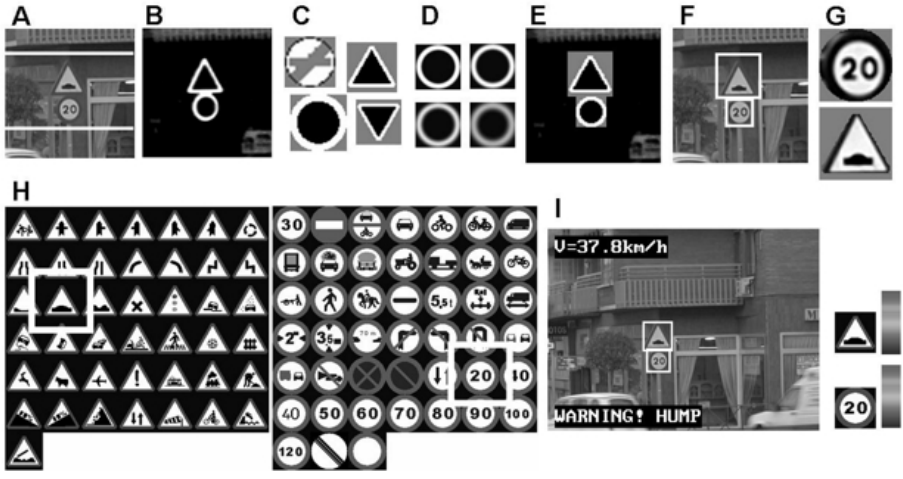


Fig. 4. A) Limited RGB image. B) HSI or RGB grayscale enhanced image. C) Some models used in correlation. D) Circular examples of Blurred Models. E) Models over the maximum score of Normalized Correlation. F) Greyscale regions to be cut for post-correlation. G) Normalized plates resized from F). H) Maximum correlation scores of G) samples over the road signs templates.I) Output: recognized signs, confidence level, speed and advices if needed.

3 Detection and Recognition

Detection is focused on danger, prohibition and yield road signs (red border), and end of limit road signs (achromatic), so two different techniques will be applied: in one hand, the vision system uses the HSI color space to enhance the red component of the road sign borders, since it values do not vary very much under lighting [14] or distance changes. What will be obtained is a grayscale image where the white color means high presence of red, and the black, absence of red (fig.4B). On the other hand, an achromatic space has been defined to isolate the end of limit road signs as they are black, grey and white. Given several images under different conditions, The plate of the road sign has been manually extracted from the background, and for all the pixels of the plate, the distance to the diagonal is obtained by the equation 1 where C_i stands for each component in the RGB colour space and I is the intensity. Data is managed statistically to have a mean value that will be the threshold that separates colour and non-colour region [fig. 5]. Applying the condition of distance over the images, a greyscale image is obtained, where white means achromicity.

$$d_i = \sqrt{\sum (C_i - I)^2} \tag{1}$$

Geometrical Limits. To improve the searching time over an image, a pinhole model and homogeneous transformation based method has been developed, where it has been considered flat world and no angle camera movement. It allows changing the

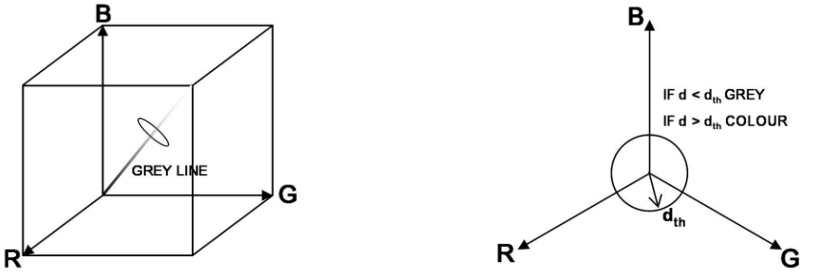


Fig. 5. A) The grey line and the limit of color and non-color in the RGB Space. B) Threshold condition of distance, seen from the grey line axis.

reference system, obtaining on the screen the position of a point in the world. Geometrical restrictions are based on traffic rules that set the dimensions of the road sign. The height gives an upper and lower limit to the scanning region, and from the width plate, distance to the road sign can be calculated. The final equation is given by:

$$v_i = \frac{V_i}{S_i} = \frac{[Zy_0 + k_y f(Y+h)]}{Z} \tag{2}$$

Where Z is the distance to the road sign, y_0 the middle screen height, k_y a conversion factor from pixel to mm, f the focal distance, Y the height of the plate and h the camera height. Given an upper and lower limit of the road sign height a restricted searching region is obtained (fig. 6).

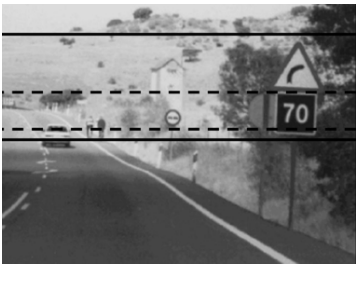


Fig. 6. An example of the geometrical restrictions: having into account the traffic rules and applying equation 2, two regions in which to look for the road signs are obtained

Due to camera movements, distance or weather conditions, the edges of the road sign borders are often fuzzy. Usually, a so-called perfect model (fig. 7) is used to correlate over the enhanced image, but this does not take into account these effects. Blurred models deal with these problems since their edges are fuzzy, so they are more realistic and obtain better scores. Different blurred models are defined in order to fulfill the majority of the situations that fuzzy the edges of the plates. As it can be seen in fig. 7, the profiles of the blurred models fits better the profiles of the borders than if a perfect model had been used.

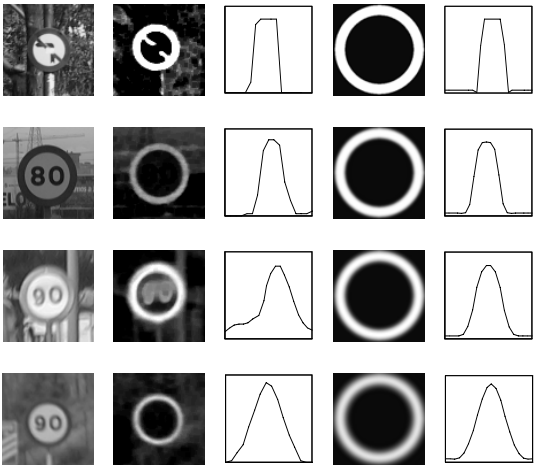


Fig. 7. From left to right: RGB images under real conditions. HSI enhanced images. Profiles of the borders. Corresponding blurred model. Profile of the blurred model. The first line corresponds to a perfect model and the others to blurred model.

Once the road sign is cut off from the grayscale image and normalized (fig. 4G), it is used as a model over the template that contains the whole road signs (fig. 4H). Normalized correlation is used again, and the output obtained with this operation gives the reliability level depicted by the coloured bar (fig. 4I).

Given the GPS speed, and having recognized a road sign, advice messages are shown in the screen (fig. 8) and acoustically emitted by the vehicle's loudspeaker in the case that the vehicle is over or below the legal limit of the recognized limit speed or end of limit road sign, or is inadequate comparing to an input user threshold in case of danger or yield road sign recognition.

4 Results

The system is able to recognize, with high reliability, prohibition, danger, yield and end of limit road signs, just as warns the driver when the speed given by a GPS is inadequate comparing to the input user limits. The achromatic space is used properly to eliminate colour from images (fig 9.A). Due to the use of HSI colour space and geometrical limits the time execution decreases (table 1). The introduction of blurred models in the detection has supposed a useful tool to improve normalized correlation scores (table 1).

Table 1. Improvements in the system attending to time execution and reliability

HSI Color Space	66% decrease in Execution time
Geometrical Restrictions	60% decrease in Execution time
Blurred Models	16.8% increase in reliability

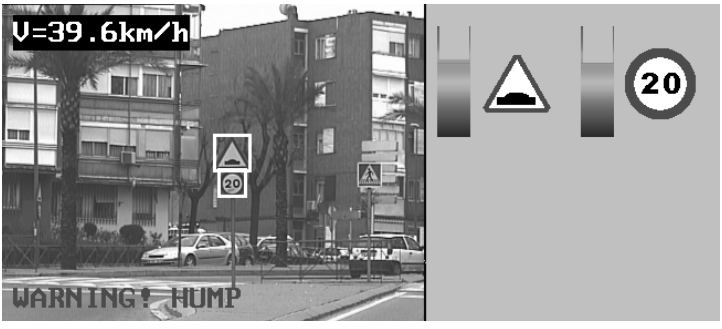


Fig. 8. Output of the system. Speed, detected road signs, recognized road signs with their reliability levels and warning message.

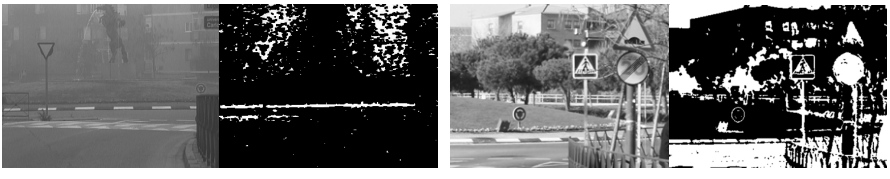


Fig. 9. A) Enhancement HSI. The road sign is over a red brick building in a foggy day. B) Binarized image obtained through application of achromatic space.

This system has been implemented and tested in the experimental platform IVVI [2] (Intelligent Vehicle based on Visual Information) (fig. 10). This vehicle allows testing the system under real conditions, which validates the results. It is equipped with two CPU's to run the algorithms and a keyboard that allows *in situ* programming. A PDA is also onboard to perform the GPS data collection and the broadcasting to the CPU's. Finally, a color camera is used in the image acquisition.



Fig. 10. From left to right: IVVI experimental platform. CPU's where algorithms run. Keyboard and control screen. Color camera, GPS and PDA.

Acknowledgments. This project is supported by the Spanish Government through the project TRA2004-07441-C03-01.

References

1. Fact sheet EURO/02/06. Copenhagen, Rome, Vienna (June 26, 2006) http://www.euro.who.int/document/mediacentre/fs0206e.pdf
2. (May 2007), http://www.uc3m.es/islab
3. Seifert, C., Paletta, L., Jeitler, A., Hödl, E., Andreu, J.-P., Luley, P., Almer, A.: Visual Object Detection for Mobile Road Sign Inventory. In: Brewster, S., Dunlop, M.D. (eds.) MobileHCI 2004. LNCS, vol. 3160, pp. 491–495. Springer, Heidelberg (2004)
4. Laumeyer, R., Retterath, J.: Patent US6363161: System for automatically generating database of objects of interest by analysis of images recorded by moving vehicle
5. Bahlmann, C., Zhu, Y., Visvanathan, R., Pellkofer, M., Koehler, T.: A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. In: Proceedings of the Intelligent Vehicles Symposium, pp. 255–260 (2005)
6. de la Escalera, A., Armingol, J.M., Pastor, J.M., Rodriguez, F.J.: Visual sign information extraction and identification by deformable models for intelligent vehicles. IEEE Transactions on Intelligent Transportation Systems 5(2), 57–68 (2004)
7. Fleyeh, H.: Color detection and segmentation for road and traffic signs. In: IEEE Conference on Cybernetics and Intelligent Systems, vol. 2, pp. 809–814 (2004)
8. Vitabile, S., Gentile, A., Siniscalchi, S.M., Sorbello, F.: Efficient Rapid Prototyping of Image and Video Processing Algorithms. In: Euromicro Symposium on Digital System Design, pp. 452–458 (2004)
9. Soetedjo, A., Yamada, K.: Traffic Sign Classification Using Ring Partitioned Method. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E 88-A(9), 2419–2426 (2005)
10. Sotelo, M.A., Rodríguez, F.J., Magdalena, L.: VIRTUOUS: Vision-Based Road Transportation for Unmanned Operation on Urban-Like Scenarios. IEEE Transactions on Intelligent Transportation Systems 5(2), 69–83 (2004)
11. Garcia-Garrido, M.A., Sotelo, M.A., Martin-Gorostiza, E.: Fast traffic sign detection and recognition under changing lighting conditions. In: Proceedings of the IEEE Intelligent Transportation Systems Conference, pp. 811–816. IEEE Computer Society Press, Los Alamitos (2006)
12. Betke, M., Makris, N.: Recognition, Resolution, and Complexity of Objects Subject to Affine Transformations. International Journal of Computer Vision 44(1), 5–40 (2001)
13. (May 2007), http://www.nmea.org
14. Chiung-Yao, F., Sei-Wang, C., Chiou-Shann, F.: Road-sign detection and tracking. IEEE Transactions on Vehicular Technology 52(5), 1329–1341 (2003)

Efficient On-Board Stereo Vision Pose Estimation*

Angel D. Sappa[1], Fadi Dornaika[2], David Gerónimo[1], and Antonio López[1]

[1] Computer Vision Center, Edifici O Campus UAB
08193 Bellaterra, Barcelona, Spain
{asappa, dgeronimo, antonio}@cvc.uab.es
[2] Institut Géographique National
94165 Saint Mandé, France
fadi.dornaika@ign.fr

Abstract. This paper presents an efficient technique for real time estimation of on-board stereo vision system pose. The whole process is performed in the Euclidean space and consists of two stages. Initially, a compact representation of the original 3D data points is computed. Then, a RANSAC based least squares approach is used for fitting a plane to the 3D road points. Fast RANSAC fitting is obtained by selecting points according to a probability distribution function that takes into account the density of points at a given depth. Finally, stereo camera position and orientation—pose—is computed relative to the road plane. The proposed technique is intended to be used on driver assistance systems for applications such as obstacle or pedestrian detection. A real time performance is reached. Experimental results on several environments and comparisons with a previous work are presented.

1 Introduction

Several vision based advanced driver assistance systems (ADAS) have been proposed in the literature during recent years (e.g., [1], [2], [3]). They can be broadly classified into two different categories: *monocular* or *stereo*. Each one of them has its own advantages and disadvantages making it difficult to decide which is the best approach for a general purpose driver assistance system.

In general, monocular vision systems avoid problems related to 3D Euclidean geometry by using the prior knowledge of the environment as an extra source of information. However, it may lead to wrong results. For instance, considering a constant camera's position and orientation is not a valid assumption to be used in urban scenarios, since both of them are easily affected by road imperfections or artifacts (e.g., rough road, speed bumpers), car's accelerations, uphill/downhill driving, among others. Facing up to this problem [4] introduces a technique for

* This work has been partially supported by the Spanish Ministry of Education and Science under project TRA2004-06702/AUT. The first author was supported by The Ramón y Cajal Program. The third author was supported by Spanish Ministry of Education and Science grant BES-2005-8864.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1183–1190, 2007.
© Springer-Verlag Berlin Heidelberg 2007

estimating vehicle's yaw, pitch and roll. Since a single camera is used, it is based on the assumption that some parts of the road have a constant width (e.g., lane markings). Similarly, [5] proposes to estimate camera's orientation by assuming that the vehicle is driven along two parallel lane markings. Unfortunately, none of these two approaches can be generalized to be used in urban scenarios, since in general lanes are not as well defined as those of highways.

The main advantage of monocular systems is their high capture rate, which is at the same time the weakest point of current stereo systems. On the other hand, the main advantage of stereo vision systems lies in the richness of 3D information, which allows to face up problems that cannot be tackled with monocular systems without having a prior knowledge of the scene. In other words, drawbacks of stereo vision systems, like the low capture rate, are related to the current technology while drawbacks of monocular vision systems are due to their monocular nature. Therefore, taking into account the fast evolution of technology it is assumed that most of the stereo systems drawbacks, which are related to the current technology, will be surpassed soon.

In this context, [6] presents an algorithm for on-board camera extrinsic parameter estimation. Although robust, the major drawback of that technique is the high CPU time required to process the whole set of data points. In the current paper a two stage technique is presented; it introduces improvements over that original approach ([6]) by using a compact set of points. An efficient RANSAC based least squares fitting approach estimates the parameters of a plane fitting to that set of points. Finally, camera's position and orientation are directly computed, referred to that plane. The proposed technique could be indistinctly used for urban or highway environments, since it is not based on a specific visual traffic feature extraction but on raw 3D data points.

The remainder of this paper is organized as follows. Section 2 presents the proposed technique. Experimental results on urban scenes are presented in Section 3 together with comparisons with previous approaches. Finally, conclusions are given in Section 4.

2 Proposed Technique

Let $D(r, c)$ be a depth map with R rows and C columns (the image size), where each array element $(r, c)(r \in [0, (R-1)]$ and $c \in [0, (C-1)])$ is a 3-vector that represents a scene point of coordinates (x, y, z) in the sensor coordinate system. Figure 1 depicts the sensor coordinate system of the stereo camera that is attached to the vehicle's windshield. Due to the orientation alignment between the sensor coordinate system and the vehicle, one can assume that vertical variations between consecutive frames—due to road imperfections, car accelerations, changes in the road slope, etc.—will mainly produce changes in camera's height and pitch angle. In other words, yaw and roll angles are not so affected by those variations. In practice, all three angles can change, however in this study we are only interested in pitch angle variations. The proposed approach consists of two stages, which are presented below.



Fig. 1. On-board stereo vision sensor with its corresponding coordinate system (right camera coordinate system is used as reference).

2.1 3D Data Point Projection and Cell Selection

The aim at this first stage is to find a compact subset of points, ζ, containing most of the road's points; similar to our previous proposal [6]. Additionally, noisy data points should be reduced as much as possible in order to avoid both a very time consuming processing and erroneous plane fits. To speed up the whole algorithm, most of the processing at this stage is performed over a 2D space.

Original 3D data points, $D(r, c)$, are mapped onto a 2D discrete representation $P(u, v)$; where $u = \lfloor D_y(r, c) \cdot \sigma \rfloor$ and $v = \lfloor D_z(r, c) \cdot \sigma \rfloor$. σ represents a scale factor defined as: $\sigma = ((R+C)/2)/((\Delta X + \Delta Y + \Delta Z)/3)$; R, C are the image's rows and columns respectively, and $(\Delta X, \Delta Y, \Delta Z)$ is the working range in 3D space—on average (34×12×50) meters. Every cell of $P(u, v)$ keeps a pointer to the original 3D data point projected onto that position, as well as a counter with the number of mapped 3D points. Figure 2*(bottom-left)* shows a 2D representation obtained after mapping the 3D cloud presented in Figure 2*(top-right)*.

Points defining the ζ subset are selected by picking up one cell per column. This selection process is based on the assumption that the road surface is the predominant geometry in the given scene—urban or highway scenarios. Hence, it picks one cell per column in the 2D projection (the cell with the largest number of points in that column). It avoids the use of a fixed threshold value for the whole 2D space. This is one of the differences with respect to [6], where a constant threshold value was used in the cell selection process.

Finally, in order to reduce the processing time, every selected cell is represented by the barycenter of its mapped points. The set of these barycenters define the sought subset of points, ζ. This data compression step is another difference with [6], where all mapped points into the selected cells were used for the fitting process. Using one single point per selected cell a considerable reduction in the CPU time is reached.

2.2 RANSAC Fitting with a Compact Set of 3D Points

The outcome of the previous stage is a compact subset of points, ζ, where most of them belong to the road (Figure 2*(bottom-right)*). However, since some outliers

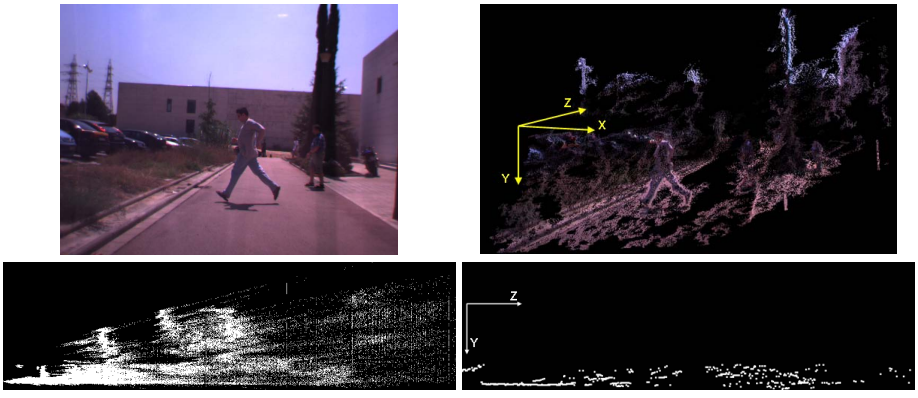


Fig. 2. (*top*) A single frame (right camera) together with the 3D data points computed with the stereo rig—notice that the image contains a large amount of holes due to occlusions and noisy regions. *(bottom-left)* YZ projection. *(bottom-right)* Cells finally selected to be used during the plane fitting stage.

are also included in that subset of points a RANSAC based [7] approach is used for computing plane parameters. Every selected cell is associated with a value that takes into account the amount of points mapped onto that position. This value will be considered as a probability density function. A cell containing a large number of mapped points, will have a high chance of being selected during the random sampling stage; at the same time RANSAC algorithm will find easier the Consensus among the whole set of point. The normalized probability density function is defined as follow:

$$f_{(i)} = \frac{n_{(i)}}{N} \tag{1}$$

where $n_{(i)}$ represents the number of points mapped onto the cell i (Figure 3($left$)) and N represents the total amount of points contained in the selected cells. Recall that we have one cell per column i. Next, a cumulative distribution function, $F_{(j)}$, is obtained as:

$$F_{(j)} = \sum_{i=0}^{j} f_{(i)} \tag{2}$$

If the values of F are randomly sampled at n points (with a uniform distribution), the application of the inverse function F^{-1} to those points leads to a set of n points that are adaptively distributed according to $f_{(i)}$. This principle is illustrated in Figure 3($right$) where three points are randomly selected.

The fitting process computes plane parameters by means of an efficient RANSAC based least squares approach. Although an automatic threshold could be computed for inliers/outliers detection, following robust estimation of standard deviation of residual errors [8], we finally decided to define a fixed value in order to reduce CPU time. Notice that robust estimation of standard deviation involves computationally expensive algorithms such as sorting function. Hence, a

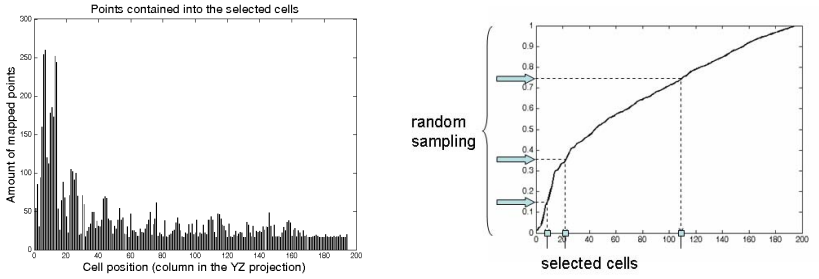


Fig. 3. *(left)* Bar diagram showing the amount of points mapped into the selected cells—recall that only one cell per column is picked up. *(right)* Cumulative distribution function computed from the amount of points mapped into every single cell.

predefined threshold value for inliers/outliers detection has been defined (a band of ±10 cm was enough for taking into account both 3D data point accuracy and road planarity). The proposed approach works as follows:

Random sampling. Repeat the following three steps K times, in our experiments K was set to 100:

1. Draw a random subsample of three different 3D points (P_1, P_2, P_3)— barycenters —where every point is drawn according to the probability density function $f_{(i)}$ using the above process (Figure 3(*right*)).
2. For this subsample, indexed by $k(k = 1, ..., K)$, compute the plane parameters[1] (a, b, c). Since P_i are barycenter points, they could define a set of collinear points; therefore, to prevent this occurs, their coordinates are set up as follow:$P_{1(x_{min}, y_{(1)}, z_{(1)})}$, $P_{2(x_{min}+(x_{max}-x_{min})/2,\ y_{(2)},\ z_{(2)})}$, $P_{3(x_{max},\ y_{(3)},\ z_{(3)})}$, where x_{min} and x_{max} correspond to the minimum and maximum x coordinate of the original whole set of points, respectively.
3. For this solution $(a, b, c)_k$, compute the number of inliers among the entire set of 3D points contained in ζ, using ±10cm as a fixed threshold value.

Solution

1. Choose the solution that has the highest number of inliers. Let $(a, b, c)_i$ be this solution.
2. Refine $(a, b, c)_i$ by using its corresponding inliers. To this end, the least squares fitting approach [9], which minimizes the square residual error $(1 - ax - by - cz)^2$ is used.
3. In case the number of inliers is smaller than 10% of the total amount of points contained in ζ, those plane parameters are discarded and the ones corresponding to the previous frame are used as the correct ones. In general, this happens when 3D road data are not correctly recovered since severe occlusion or other external factor appears.

[1] Notice that the general expression $ax + by + cz + d = 0$ has been simplified dividing by $(-d)$, since we already known that $(d \neq 0)$.



Fig. 4. Vanishing lines computed according to the current camera pose—camera height and pitch angle

Finally, camera's height (h) and orientation (Θ), referred to the fitted plane (a, b, c), are easily computed. Camera's height is given by: $h = 1/\sqrt{a^2 + b^2 + c^2}$. Camera's orientation—pitch angle—is directly computed from the current plane orientation: $\Theta = \arctan(c/b)$. Both values can be represented as a single one by means of the vanishing line (e.g., [10], [11]). The vanishing line position (v_i) for a given frame (i) is computed by back-projecting into the image plane a point lying over the plane, far away from the camera reference frame, $P_{(i)}(x, y, z)$. Let ($y_{(i)} = (1 - cz_{(i)})/b$) be the y coordinate of $P_{(i)}$ by assuming $x_{(i)} = 0$. The corresponding $y_{(i)}$ back-projection into the image plane, which define the row position of the sought vanishing line, is obtained as $v_{(i)} = v_{(0)} + fy_{(i)}/z_{(i)} = v_{(0)} + f/z_{(i)}b - fc/b$; where, f denotes the focal length in pixels; $v_{(0)}$ represents the vertical coordinate of the principal point; and $z_{(i)}$ is the depth value of $P_{(i)}$ (in the experiments $z_{(i)} = 10000$).

3 Experimental Results and Comparisons

The proposed technique has been tested on different urban environments and compared with [6]. A 3.2 GHz Pentium IV PC with a non-optimized C++ code was used. The proposed algorithm took, on average, 90 ms per frame including both 3D points computation and on-board pose estimation. Notice that this is about four times faster than our previous approach [6], while the same results are obtained.

Figure 4 shows two different frames with their corresponding vanishing lines computed with the proposed technique. The computed camera height and pitch angle, as a function of time, for this sequence are presented in Figure 5. Both values are referred to the current fitted plane. This sequence contains a gentle downhill, vehicle's accelerations and two speed bumps. As can be seen, neglecting these variations will affect further processing (e.g., car or pedestrian detection, collision avoidance, etc.).

Finally, Figure 6 presents results obtained after processing a 12 second video sequence corresponding to a short flat road followed by an uphill (10 fps are depicted). Notice how the pitch angle changes during the sequence according

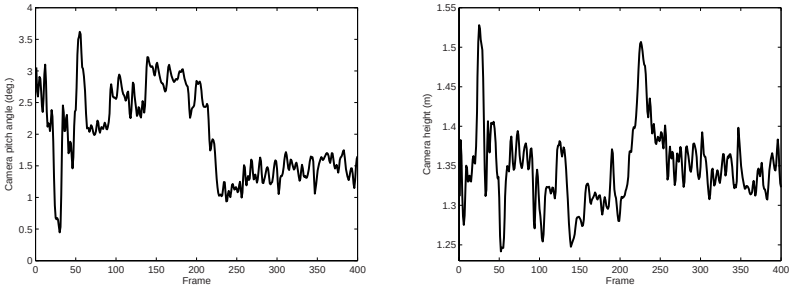


Fig. 5. *(left)* Camera pitch angle for the video sequence of Figure 4 (only 2 fps are plotted). *(right)* The corresponding camera distance to the fitted plane at every frame.

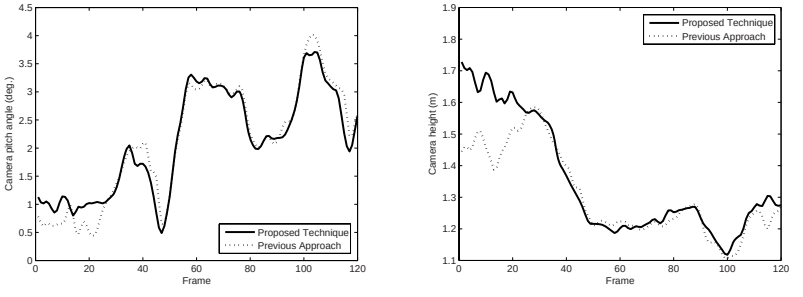


Fig. 6. Comparisons between the proposed technique and [6] with a 12 second-long video sequence: *(left)* Camera pitch angle. *(right)* The corresponding camera height to the fitted plane at the every frame.

to the current geometry. In this scene there are no speed bumps and the car keeps almost a constant speed after the initial acceleration, used for starting the car's motion. In both plots, the obtained results are presented together with the results obtained with [6]. As can be appreciated, although the obtained results have similar trend, the new proposed approach behaves better than the previous proposal in those critical situations where two different geometries converge, first 40 frames—in this case a flat road with a quite sharp uphill. Since our previous proposal uses a constant threshold value for cell selection (Section 2.1), only cells near to the sensor were considered; on the contrary, with the new approach all candidate cells are considered

4 Conclusions

An efficient technique for a real time pose estimation of on-board camera has been presented. The input data are a set of 3D points provided by the on-board stereo camera. After an initial mapping a compact set of 3D points is chosen as

candidate for fitting a plane to the road. The RANSAC technique selects points according to a probability distribution function that takes into account density of points at a given position. Although it has been tested on urban environments, it could be also useful on highway scenarios. A considerable reduction in the CPU processing time was reached by working with a reduced set of points selected according to a continuously updated probability distribution function. The latter drives to a faster convergence during the RANSAC fitting stage.

References

1. Bertozzi, M., Binelli, E., Broggi, A., Del Rose, M.: Stereo vision-based approaches for pedestrian detection. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, San Diego, USA, IEEE Computer Society Press, Los Alamitos (2005)
2. Gerónimo, D., Sappa, A., López, A., Ponsa, D.: Adaptive image sampling and windows classification for on–board pedestrian detection. In: Proceedings of the International Conference on Computer Vision Systems, Bielefeld, Germany (2007)
3. Hautière, N., Tarel, J., Lavenant, J., Aubert, D.: Automatic fog detection and estimation of visibility distance through use of an onboard camera. Machine Vision and Applications 17(1), 8–20 (2006)
4. Coulombeau, P., Laurgeau, C.: Vehicle yaw, pitch, roll and 3D lane shape recovery by vision. In: Proc. IEEE Intelligent Vehicles Symposium, Versailles, France, pp. 619–625. IEEE Computer Society Press, Los Alamitos (2002)
5. Liang, Y., Tyan, H., Liao, H., Chen, S.: Stabilizing image sequences taken by the camcorder mounted on a moving vehicle. In: Proc. IEEE Int. Conf. on Intelligent Transportation Systems, Shangai, China, pp. 90–95. IEEE Computer Society Press, Los Alamitos (2003)
6. Sappa, A., Gerónimo, D., Dornaika, F., López, A.: On-board camera extrinsic parameter estimation. Electronics Letters 42(13), 745–747 (2006)
7. Fischler, M., Bolles, R.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Graphics and Image Processing 24(6), 381–395 (1981)
8. Rousseeuw, P., Leroy, A.: Robust Regression and Outlier Detection. John Wiley & Sons, New York (1987)
9. Wang, C., Tanahashi, H., Hirayu, H., Niwa, Y., Yamamoto, K.: Comparison of local plane fitting methods for range data. In: Proc. IEEE Computer Vision and Pattern Recognition, Hawaii, pp. 663–669. IEEE Computer Society Press, Los Alamitos (2001)
10. Zhaoxue, C., Pengfei, S.: Efficient method for camera calibration in traffic scenes. Electronics Letters 40(6), 368–369 (2004)
11. Rasmussen, C.: Grouping dominant orientations for ill-structured road following. In: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, pp. 470–477. IEEE Computer Society Press, Washington, USA (2004)

Complex Software Problem Solving by Means of Abstractive Techniques

David Davis

ICT Group, Faculty of Engineerin
University of Technology, Sydney, Australia
david.davis@uts.edu.au

Abstract. Studies have shown that a large number of medium to large scale software intensive system development are cancelled. The primary reasons include the loss, lack or mis-understanding the problem to be "solved" – from customer to engineers and even amongst the engineers themselves. The Object Boundary/Context modelling process is key to immediate "system" understanding of the problem space at the highest level, enabling a "context" for formulating plans and requirements elicitation. Furthermore, it provides a communication point between all stakeholders and enables the correction of misunderstandings at the earlier stages of a software intensive system development. The model does not seek to supplant conventional formal analysis/architecture/design but acts as the earliest model to ensure that complex software problems do not turn "wicked".

1 Introduction

This paper argues that the first approach to any "complex" software development is the representation of the problem space in an engineering shape grammar, reasoned by using abstractive techniques as embodied by the Socratic method and realised using a Boundary/Context model. This is predicated on the notion that it is unsound, misleading, and ineffective to question or reason about the cause of a phenomenon before verifying its existence and understanding its essence. There are three major classifications of problems: Simple: having usually one solution; Complex: having many solutions and Wicked: having no solutions until the problem is "tamed" [1]. This paper focuses on Complex software problems.

Bertrand Russell once said: "... Although this may seem a paradox, all exact science is dominated by the idea of approximation..." [2]. Many would argue that software, per se, is hardly an exact science, given the number of ways a software problem space could be analysed, designed and implemented. But modern software engineering craves the certitude of minimal risk implementation to ensure an outcome, even a satisfactory outcome, given the appalling figures based on statistics drawn from more than 6,000 corporate and government projects, that about 48% of 10,000 function points and 65% of more than 100,000 function point projects were cancelled [3].

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1191–1198, 2007.
© Springer-Verlag Berlin Heidelberg 2007

After 30 years as a software engineer and a further 5 years as a tertiary educator in the same field, it has become apparent that there is massive confusion between the two non-trivial software development qualities: process and product. Software Engineering is often viewed as being process driven and project failures can often be attributed to excessive process, with minimal regard to the "problem space" – the problem itself. Equally, undue emphasis on the problem space with minimal regard to process can also lead to project failure. Thus, there is a mutually inclusive relationship between process and product best exemplified by the "manufacturing metaphor" with employed processes sufficient only to support the development of the product which conforms to an agreed on quality and nothing more [4].

2 The Profession of Software Engineering

McConnell [4] argues there should be greater stress on engineering, rather than on the direct software coding, in a major or large scale Software Intensive System Development. This implies there should be a stronger emphasis on how the project is performed, than the inevitable coding and this is in keeping with aims of the original 1968 NATO conference set up to counter the so-called software crisis at the time and coined the term Software Engineering, with emphasis on the Engineering of a Software Intensive Project [5].

Peter Naur et al coined the term "software engineering" at NATO conference in Garmisch-Partenkirchen and pointed out software should following an engineering paradigm, it was the response to software crisis which provided with too little quality, too late in delivering and over budget [6]

McConnell [4] also argues that elements, which contribute to the success of major or large scale Software Intensive System Developments, include:

- Discipline.
- Understanding simplicity.
- Understanding the problem (to be solved by the project).
- Using standards and processes that have been around for over 30 years or more.

3 Software Intensive System Development Models

Based on the working party's "construction metaphor" for the IEEE Standard 1471, 2000, "Architectural Descriptions for Software Intensive Systems" [7], a focus for discipline is the process control mechanism that includes:

- Development Lifecycles,
- Process models – e.g. the linear sequential or iterative development model.
- Development paradigms, such as the Object or Structured.
- Architecture models.
- Build or integration models and strategies.

– Testing methodologies such as testing to code or specification.
– Validation.

M. Kramer, states that "... decades of work in experimental psychology have revealed that when people are free to choose between a simple visual image and a more complex one, they gravitate to the more complex one..." [8]. This implies humans may prefer complexity over simplicity. Often good project personnel fail and consequently projects fail, because those personnel don't understand the difference between simplicity and complexity.

4 Problem Types

There are three classes of problems encountered or solved by engineering projects:

– Simple, which has only one solution.
– Complex, which has many solutions.
– Wicked [1] which has no solutions.

It has always been my contention that failed projects of the Capers-Jones ilk [3], has an initial problem statement that is "complex" to the point of hovering on the edge of "wickedness". This boundary condition is alas often not realized until some time later in the project, often when it is too late, too late. To forestall this potential of the complex problem tumbling into the well of wickedness, the problem needs "taming". Then using the Rittell [1] method, apply analytical models for further development. It is a well known axiom that it is unsound, misleading, and ineffective to question or reason about the cause of a phenomenon before verifying its existence and understanding its essence. Yet in a large number of the studies made by Capers-Jones [3] this is what happened:

– Customer's version of the problem space was different from that of the Engineers, and
– Even among the Engineers there were often major differences in understanding the same problem.

An effective initial resolution mechanism is to then depict candidate problem solutions as a simple model utilising an easily understood Engineering Shape Grammar, for discussion and understanding by all project stakeholders.

5 Metaphors and Models

Software Engineering is full of metaphors and models – in fact an entire, non-trivial software development project is a consistent "chain" of models where each model, as Simon Bennett, et al, pointed out: "A model represents something else – it is a metaphor/analogy that explains a phenomena more simply, by associating it with explanation of something else ... provides a complete view of a system at a particular stage and from a particular perspective... A useful model has just the right amount of detail and structure and represents only what is important for the task at hand..." [9].

6 Object Boundary Model and Its Rules

A typical Engineering Shape Grammar for providing the initial model is the Boundary/Context Model which has been an engineering tool used by most branches of engineering for at least 30 years (principally Electrical and Hydraulic) [10]. It was an approximation to understand the behaviour of an engineering system at earlier stages in its development, and migrated to Software Engineering as a part of the Structured Paradigm, where it was extended to show data linkages between blocks of logic [11]. The key feature of this method is its capacity to show different layers of a problem, in a top down hierarchy, from the most abstractive to the least.

In graduating to the Object Paradigm certain changes are made to form it into the Object Boundary/Context Model. These include rules such as:

- A functional abstractive (object/noun) approach rather than the traditional behavioural (verb) approach.
- Employs the Socratic method [12], reason and logic for determination of the model's objects and their relationships – principally:
  - Divergent/Convergent questioning [13].
  - Inductive/Deductive reasoning.
- Object oriented rules such as encapsulation and re-use [9].
- UML-like nomenclature for it's engineering shape grammar [14].
- Verified by design criteria including Functional Cohesion and Normal Coupling [15].

When these rules are applied, the Object Boundary/Context model becomes a powerful tool for a first approximation of a software intensive system development's problem space.

The Socratic method, as translated for complex software problem solving, is best described as: abstracting problems to the point of understanding (major functional abstractions or major components) and then increase complexity in a series of logical steps [16]. By understanding the problem, you understand its candidate solutions.

The Socratic method is often accompanied by Seneca's principle: Balance optimism with pessimism (the stoic) [17], which engineers translate as expecting risk; and Occam's razor: When confronted by candidate solutions, the most likely correct one is that, that is both logical and most simplest [18].

In convergent/divergent questioning, the key distinctions: convergent questions operate in the knowledge domain, questioner attempts to converge on and reveal "facts" and answers have truth value; and divergent questions operate in the concept domain [13]. It is an effective inquiry for problem solving where the convergent component - builds up to asking deep reasoning questions by systematically asking lower-level, convergent questions, and the divergent component is where generative questions are asked to create concepts on which the convergent component can act.

6.1 Engineering Shape Grammar Nomenclature

In applying the Engineering Shape Grammar for the Object Boundary/Context model, the problem space is represented by a rectangle with rounded corners as shown at Figure 1. Inputs/Outputs to the problem space are represented by Jacobson actors [14].

Major functional abstractions are shown as "objects" (named minor rectangles) within the problem space and are constructed using traditional "black box" methods pioneered by Meyers [19] and others and represent high level functional or sequential cohesions (major system components).

Actors and objects are control coupled using normal coupling nomenclature [14]. Hence, traditional design rules of maximum cohesion and minimal coupling, coupled with logical reasoning, are a quality measure of the completed model.

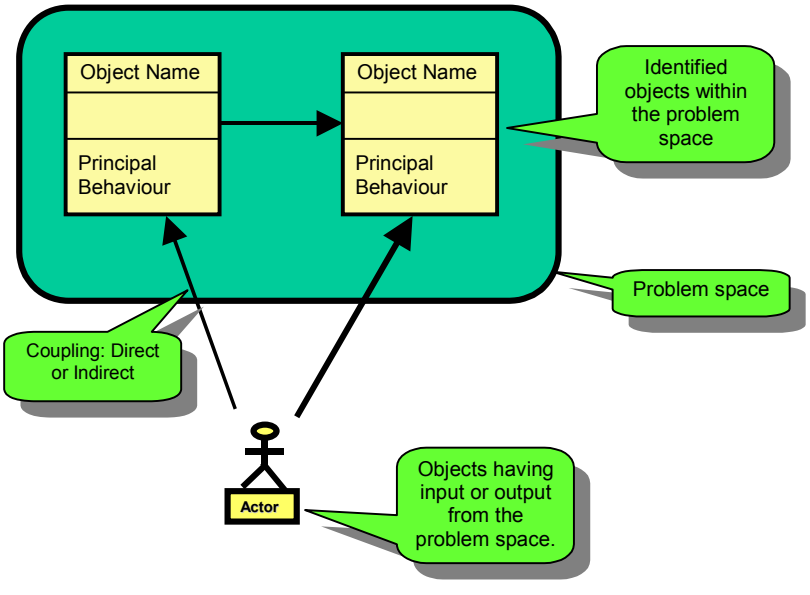


Fig. 1. Object Boundary/Context Model Nomenclature

6.2 Model Uses and Interpretation

The Object Boundary/Context Model, as a Boundary Model, it provides a control mechanism for the construction of the three Software development plans: Software Development Plan (SDP), Software Test Plan and Software Quality Plan, by maximising understanding of the problem space and the early identification of major system risks within the development project. As well as empowering the stakeholders, including the customer, with an earlier understanding of the project, the 4 to 1 accuracy problems of project size estimation can be reduced to as low as 2 to 1.

It permits extensibility modelling at an early stage of the project and assists in process model selection - although complex software projects generally employ the iterative/incremental process model which the model assists by the identifying the system "core" and using extensibility modelling identifies functional layering controlled by Lehmen's 5th Law [20].

As a Context Model, it is used as an elicitation mechanism for Requirements garnering. The model identifies principal functional and behavioural abstractions, constraints, modes of system operation, basic system qualities including performance and security; system "responsibility" abstractions such as persistence, business logic and presentations. The use as an elicitation mechanism focuses the analyst on the problem space by providing the underlying system metaphor as a framework, which minimises derived and redundant requirements. A structure of a Software Requirements Specification (SRS) using this methodology is the subject for another a paper.

6.3 Case Study: The Library Problem Structure

When applied to a generalised library problem as shown at Figure 2, the Boundary/Context model comprises two actors: Users and Staff. The problem space accommodates two objects: Items and Accounts. The User actor is directly coupled to Items and indirectly coupled to object: Accounts, via Items. Staff actor is directly coupled to objects: Items and Accounts. Object: Items is directly coupled to object: Accounts.

When examining the "accuracy" of the model, inductive reasoning shows the most likely software system relationship is: Object User has a direct relationship with object Items (showing borrowed and returned behaviour by object User) but an indirect relationship with object Accounts through object Items (the user is charged membership fees to borrow items and late fees could be imposed); object User has an indirect relationship with Staff (through Items – Staff being secondary to Items) and object Staff updates/controls the persistence of object Items as well as being paid and/or other Account controls.

6.4 Case Study: The Library Problem Interpretation of the Model

The Library Boundary/Context model, as depicted at Figure 2, identifies four (4) major functional abstractions: User, Staff, Items (class of objects including books, magazines, etc.) and Accounts; six (6) behavioural abstractions: User to Items, User to Accounts, Staff to Items, Staff to Accounts, User to Staff and Items to Accounts; system qualities including: Performance, Reliability, Security and (possibly) Maintainability; and System responsibilities including: Persistence, Presentation (User Interfaces, minimum three) and Business Logic. Above all it identifies a concurrent system (User and Staff modes), possibly identifies a Client/Server Architecture and exposes a potential minimum risk system development framework which is a "point of sale".

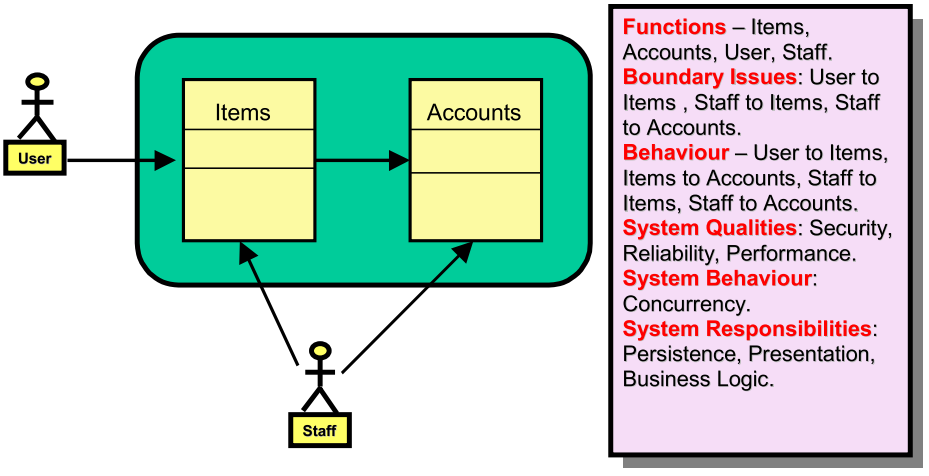


Fig. 2. Object Boundary/Context Model Applied to the Library Problem

6.5 Conclusion

This approach to abstractive techniques is not new. It uses a philosophical methodology that has been in existence for over 2000 years. The methodology is simple and complete. Best of all it uses traditional methods of convergent/divergent questioning and inductive/deductive reasoning as its control mechanisms. But like most modelling techniques, while the methodology is simple and the depiction in the engineering shape grammar straight forward, "simple aint simple". As Richard Feynman once opined: "...It is often easier to learn about the modelling substrate (probability theory, simulation methodology, etc.) than about the modelling process itself..."

To learn modelling is mostly a matter of experience and practice but always requires a thorough understanding of the systems to be modelled.

Nonetheless, the abstractive techniques discussed by this paper are a powerful way to represent an initial approximation to a complex software problem space and provide an analytical guidance mechanism for minimising risk within a software development project thus decreasing the likelihood of project failure represented by the statistics of Capers Jones. Further, by understanding the problem in the Socratic meaning, the application of process can be focussed on the problem and allows the rational use of agile technologies and "just in time" deployment of development personnel.

References

1. Rittell, H., Webber, M.: Dilemmas in a general theory of planning. Policy Sciences 4 (1973)
2. Auden, W.H., Kronenberger, L.: The Viking Book of Aphorisms: A Personal Selection. New York Dorset Press (1981)

3. Jones, C.: Applied Software Measurement: Assuring Productivity and Quality, 2nd edn. McGraw-Hill, New York (1996)
4. McConnell, S.: After the Goldrush: Creating a True Profession of Software Engineering. Microsoft Press, Redmond, Washington (1999)
5. Randell, B.: Memories of the NATO Software Engineering Conferences, http://www.cs.ncl.ac.uk/research/pubs/articles/papers/174.pdf
6. http://www.comphist.org/computing_history/new_page_13.htm
7. Maier, M., Emery, D., Hilliard, R.: Software Architecture: Introducing IEEE Standard 1471. Computer, IEEE Computer Society 34(4), 107–109 (2001)
8. Kramer, M.: Making Sense of Wine. Running Press Book (September 2003)
9. Bennett, S., McRobb, S., Farmer, R.: Object-Oriented Systems Analysis and Design: Using UML, 3rd edn. McGraw-Hill, New York (2006)
10. Hyman, B.: Fundamentals of Engineering Design, 2nd edn. Prentice-Hall, Englewood Cliffs (2003)
11. De Marco, T.: Structured Analysis and System Specification. Yourdon Press Computing Series (1979)
12. http://www.garlikov.com/Soc_Meth.html
13. Dym, C., Agogino, A., Eris, O., Frey, D., Leifer, L.: Engineering Design Thinking, Teaching, and Learning. Journal of Engineering Teaching and Learning, 103–120 (2005)
14. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modelling Language Manual, 2nd edn. Addison-Wesley, London, UK (2005)
15. Page-Jones, M.: The Practical Guide to Structured Systems Designs, 2nd edn. Prentice-Hall, Englewood Cliffs (1988)
16. Halford, G., S.: Children's Understanding: The Development of Mental Models. Lawrence Erlbaum Associates, Mahwah (1993)
17. http://www.wsu.edu/~dee/ROME/IMPROME.HTM
18. Jerrold, K.: Realistic Rationalism. MIT Press, Cambridge (1998)
19. Meyer, B.: Object-Oriented Software Construction, 1st edn. Prentice-Hall, Englewood Cliffs (1988)
20. Sommerville, I.: Software Engineering, 7th edn. Addison-Wesley, London, UK (2004)

Active MIB: Addressing Challenges of Wireless Mesh Networks

Zenon Chaczko, Frank Chiang, and Robin Braun

ICT Group, Faculty of Engineering
University of Technology, Sydney, Australia
zenon@eng.uts.edu.au, frankj@eng.uts.edu.au, robin.braun@eng.uts.edu.au

Abstract. There is a list of very much needed object-oriented and autonomous paradigms that are currently absent from traditional implementations of MIBs in telecommunication systems. We argue that some of autonomous features can be inherent in an application of an Active MIB mechanism itself, and thus this type of application would be a more appropriate substrate in which to model, develop and integrate the benefits brought by the object-oriented approach. In our work, we claim that active networks are such an application area, in which the object-oriented and autonomous system ideas can be successfully applied. The objective of this research paper is to demonstrate applicability and compatibility of the AMIB concept with functional and architectural requirements of wireless mesh networks. This paper introduces new approaches for using AMIB-based mechanisms in telecommunication networks as well as discusses autonomic processes and algorithms that can be used to manage such networks.

1 Introduction to Wireless Mesh Networks

In recent years, there is much discussion about the Next Generation Networks (NGNs) initiatives [6, 10, 18]. Many researchers see the NGSs as being highly dynamic and Service Orientated. There are many voices that usefulness of future NGNs would not depend as much on number of networks nodes but on the quality of connections between them [1, 2]. The quality of connections directly depends on how interconnected nodes meet the end-users' ever changing demands for these networks to be highly adaptable, self-healing and reconfigurable while maintaining efficiency, robustness, security and economic-effectiveness. Within the research and practice of NGNs domain, Wireless Mesh Networks (WMNs) will facilitate their efficient, flexible and secure deployment as well as their discrete mobility. The WMNs concepts are not based on new ideas, however there are requirements that these networks cannot have any centralized control, that they can be massively distributed, and that they are capable of providing constant and survivable connectivity together with value-added services.

From a system perspective, no effective autonomic management framework has ever been deployed. Multiple constraints limit throughput capacity and fairness with key issues such as: power consumption, mobility levels, radio channel

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1199–1207, 2007.
© Springer-Verlag Berlin Heidelberg 2007

allocation or interferences, QoS parameters (e.g., dynamic topology control, balance backhaul connections). We argue that autonomic management paradigm can be the solution to scalability and QoS requirements for distributed WMNs. Our goal is to address the unsolved scalability issues for WMNs by creating new routing protocols based on resource orientated cost function, topped with an efficient self-managed network management paradigm [12, 13]. We aim to experiment on all of these via multimedia streaming, testing QoS for multi-hop wireless networks. We believe that our 4/3 holonic management structure, combined with the notions of an OO AMIB and biomimetic adaptation strategies will facilitate a working solution (see Fig. 1)

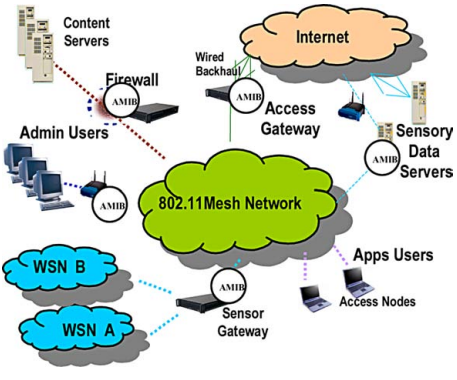


Fig. 1. WMS with fixed access points/mesh routers and mobile clients with wired/wireless connections for Internet backbone access

WMN are rapidly becoming a part of the overall business and public network infrastructure. Yet, suitable interconnectivity standards for mesh networks are still being under development [12, 13]. Such situation creates serious difficulties and challenges in the network management domain. The major challenge is an ability to integrate mesh networks so they could work in robust and reliable manner. Other challenges can be grouped into the following categories: Management of Mesh Networks; Quality of Service (QoS), roaming and technological interoperability (including wireless technologies) [5].

Managing a limited number of user applications is usually not that hard, but more often these days various businesses install increasing number of them. This requires significant efforts to be spent on updating of such infrastructure components as: firmware, policies updates, authentication keys and configuration upgrades. Business application often allow for remote updating of these components. Update activities are supported by software tools that can access the applications settings typically through a Management Information Base (MIB). These software tools utilise the Simple Network Management Protocol (SNMP) to perform queries by management stations on MIBs of managed objects (i.e. routers, gateways, or switches). Components controlled by the management sta-

tion need a SNMP software agent that can communicate with the SNMP manager. SNMP uses a specified set of commands and queries. The MIB contain capability specifications that are commonly used for wired gateways, routers, multiplexers and switches with a view to tuning the network transport to the current needs. MIBs are periodically updated to add new functionality, remove ambiguities and to fix defects.

Wireless network components; however require additional configuration parameters related to managing the radio strength or power requirements for which there are no standard MIBs available. The 802.11 standard put the power in the end nodes, which places a burden on the task of performing handoff as more control is needed by the intermediate devices that don't have such control. [17]. To handle this problem, users have to standardise on installation of their wireless applications, gateways, switches, and routers from one vendor and then rely on that particular vendor's proprietary management tools or use a solution from a third-party tool provider- that manages multiple vendors' hardware.

2 Management Information Base

A Management Information Base (MIB) is a database structure which contains a formal description of a set of network objects (entities/ components) that can be managed using the Simple Network Management Protocol (SNMP). In general, MIB contains a directory listing information on configuration settings and performance statistics pertaining a specific entity or device (eg. a router, a switch, or a multiplexer). The MIB data structure is a schema that is conceptually organised as a tree of objects and it contains the hierarchical order of all of the managed objects. Entries in MIBs are addressed through unique object identifiers (OIDs). The identifier contains the type (i.e. counter, string, gauge, or address), access level (i.e. read/write), size limits (restrictions), and object's range information. Standardized MIB formats allow SNMP based tools to monitor any device defined by a MIB.

Whilst wireless management devices use SNMP MIBs to manage hardware, standards for wireless-specific MIBs could be even more useful for measuring utilisation, channel switching and managing RF output power levels. However, for generation of such information new standards are needed. These standards have to be more flexible to allow adaptation for specific uses. Also, there are requirements to get settings, policies, and parameters from the individual client object, which at present doesn't contribute any information to the access point to help manage the RF signals. There is a debate on how far these standards should extend. As switches in WSN gain wider acceptance, various users would require support for lightweight applications, which are managed in groups by controllers in a hierarchy of nodes and branches. This situation could reduce the complexity and cost of applications but can lead to reduced interoperability among vendors' hardware. At present wireless application are interconnected to one another and to switches and routers through the wired LAN, so management-related backhaul data is carried through the higher-capacity, more-secure LAN.

As mesh networks are more broadly deployed, traffic management might become a serious issue. Typical applications can handle a several connections at a time, a dynamic nature of data traffic means that most business applications may not handle that type of traffic easily, especially if their applications cannot balance the traffic to one another during peak demand. An increase in applications for warehousing and hospitality industries may lead to potential saturation and security threats. Bandwidth saturation could occur where large groups of users might suddenly appear, such as at hotspots, requiring both prioritisation and handoff to other applications. Bandwidth issues could appear as various organisations deploy VoIP on the WLAN for mobile workers, such as within a university or corporate campus to allow follow-me-anywhere IP-based telephony systems that permit both wired and wireless access.

3 Active MIB

Active MIB reflects three abstractions (representations) of an object in a mesh network: :structure, behaviour and topology (geometry) The structural representation reflects traditional MIB data, object behavioural representation reflects objects activities and their intensity in time and topological representation reflects the object's location in and geometry relationship with the network. Behavioural dimension of device should include the interference model data which then could be used for selection of overlapping channels and calculation of radio leakages beyond just one hop only. Also, this data can be used for optimisation strategies that consider load factors, maximise number of concurrent transmissions and minimise interference factors. In Mesh Networks, the topology is often perceived to a constraint as most current models tend to preserve the original topology.

In mesh networks the knowledge of the neighbours' location is critical for directional nodes. Location estimation method often requires expensive hardware

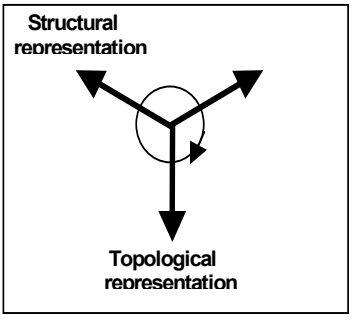


Fig. 2. Dimensions of Active MIB

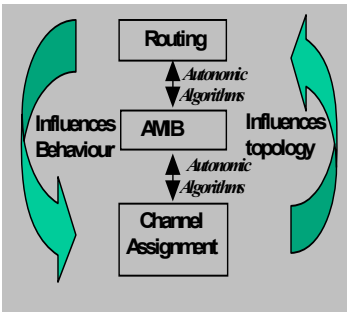


Fig. 3. Active networking using Active MIB

such as GPS and there are physical limits associated with this technology. Calculation of Angle of Arrival (AoA) and Direction of Arrival (DoA). Each node can cache estimated Angle of Arrival (AoA) data from neighboring nodes whenever it hears any signal, disrespecting the fact whether the signal is sent to it or not. When a given node has data to send, first it searches for prestored AoA information in its AMIB, if the AoA is found, the node will send a directional RTS, otherwise, the RTS is sent in omni-directional mode. The nodes in Mesh Networks update their AOA information each time they receive newer signals from the same neighbours. Data stored in AMIB is then invalidated when a node fails to obtain the CTS after 4 directional RTS transmission attempts. Additionally, the AMIB can contain both the informed discovery data: An AoA, a location table with multi-hop routing location information, the non-informed discovery (ad-hoc discovery) as well as a synchronized TR-BF (Transmission Reception Beam Forming).

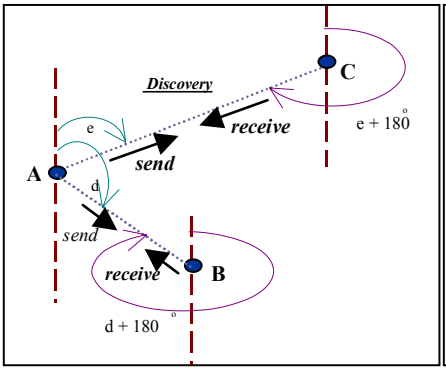


Fig. 4. Calculation of AoA's

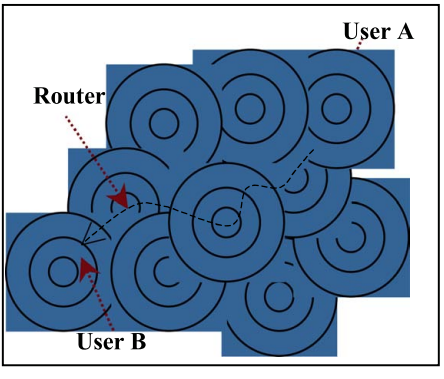


Fig. 5. Router and user geolocation

AMIB can support a geolocation mechanism. This can be done by storing measured ranges between mobile users and some known fixed points (i.e. gateways, routers or switches). Similarly to cellular systems, a triangulation method could be utilised with much better precision as "cells" are often much smaller. Many other improvements can be made as users can talk to each other.

There are other facilities the AMIB could support as well. These could be: load balancing mechanism with QoS capabilities, multimedia services, route discovery and rediscovery mechanism with optimization criteria (i.e. minimum hops or delays, maximum data rates, minimum error rates, minimum RTA or power consumption and many other parameters).

Active networks is a model of communication that allows injection of customised programs into network's nodes to dynamically change the operation of the network. The architecture of an active network fundamentally differs from that of traditional networks in which architectural qualities such as security, robustness or stability are enforced by removal of operational complexities and

suppression of capability of network components to alter its basic functions from underlying network. Active networking can enable mechanisms for highly specialised, rapid "real-time" modifications to operations in the underlying network supporting capability of sending or activating resident mobile code along with packets of information and thus allowing the data to modify its form to match the characteristics of radio channels. The tiniest program that can generate a sequence of data should adhere to the definition of descriptive or Kolmogorov Complexity principle [8, 9]. In order to compose autonomous network services, various applications of real-time Genetic Algorithms (GA), Artificial Neural Networks (ANN) or Artificial Immune Systems (AIS) based processes and algorithms can be also enabled within the active networking.

For example, an AMIB can contain the GA algorithm which calculates the multivariable h-MOP function where concludes the optimal path by considering the equation (1) as follows:

$$\begin{aligned} C_{E_{ij}}(t) &= f(Bw(i,j,t), Delay(i,j,t), Capacity(i,j,t), Cost(i,j,t)) \\ &= \left(\frac{Bw_{i,j}(t)}{\sum_{i=0,\ j=0}^{n} Bw_{i,j}(t)} \right)^{\lambda} + \left(\frac{\tau_{i,j}(t)}{\mathcal{C}_d(t)} \right)^{\lambda} \\ &\quad + \left(\frac{C_{i,j}(t)}{\sum_{i=0,\ j=0}^{n} C_{i,j}(t)} \right)^{\lambda} + \left(\frac{Cost_{i,j}(t)}{\sum_{i=0,\ j=0}^{n} Cost_{i,j}(t)} \right)^{\lambda} \end{aligned} \tag{1}$$

The proposed Holonic MIBs have the following characteristics: Firstly it is object-oriented MIBs with methods embedded. Secondly, this MIB exists in holonic-level. Holonic is a term initiated from the manufacturing industry and later applied widely in robotics. The holonic MIBs is embedded into individual vehicle, individual electronic devices like mobile phone set, printer; and even further into sublevel of devices such as chip-level (See Figure 6). Applying this holonic concepts into MIBs are inspired from the hot researches in intelligent robotics corporation (IRC) where holonic technology is embedded in individual electronic devices. They will have the capability to communicate with each other and enable a Holonic Intelligent System in which subsystems can coordinate, cooperate, and collaborate, these are suitable for hierarchical network infrastructure. As being inspired from those ideas, we believe those holonic MIBs should also communication, collaborate with each other. The methods designed in Holonic MIBs can help to invoke this communication between two HMIBs.

Structure of Holonic MIBs consists of three parts: (1) conventional MIB ; (2) user-accessible provisioning, (3) methods/action.

When Holonic Technology is embedded into individual electronic devices, they will have the capabilities to communicate wirelessly with each other and enable

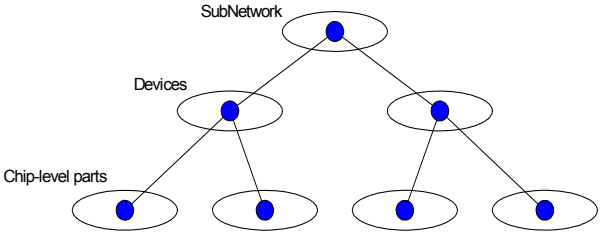


Fig. 6. Holarchy of Holonic Components

a Holonic Intelligent System in which subsystems can coordinate, cooperate, and collaborate. The intelligent components in holonic system obey advanced bi-lateral communication patterns evolved from auction and game theory, implemented with advanced peer-to-peer (P2P) algorithms, which have core features akin to popular P2P Internet utilities. The use of this intelligent component can both determines which holonic node having a better opportunity to achieve system-wide goals or local goals, and determines the most efficient and secure way to send data, and relocating digital services to more opportunistic nodes in order to improve efficiency. In addition, Holonic Technology has the desirable scalability to augment itself through our proprietary agent architecture so that subsystems can be upgraded, extended, and enhanced without shutting down.

4 Conclusion and Future Work

Support of effective management and QoS in WMN requires a new type of instruments and support mechanisms operating at various holonic levels. A load balancing mechanism, for example, could allow effective control of available resources. There is a need to support business models based on QoS differentiation to allow increase service experience to users. Traditional approaches to load balancing rely on standard WLAN technologies mechanism which reuses previous performances results at MAC and PHY layers. However, implementations must be adapted to different cases (i.e. enforcement of load transfer, holonic collaborative scenarios etc.).

Methods associated with an AMIB object are pieces of software which have privileged access to it's state. Testing if a given AMIB object has or doesn't have certain properties that are "derived" from its state could be efficiently executed by calling a dedicated method rather than by running queries on the attributes. One of the most important characteristics of object oriented AMIB would be the fact the commonality between the network management application and the AMIB's implementation preserves the strong typing mechanism and policies in the communication between components at the various holonic levels (layers).

Policies refers to a generic name for a type of code that can be automatically activated when certain events occur. These policies can be perceived as a form of constraint programming.

References

1. Akyildiz, I., Wang, X.: A survey on wireless mesh networks. IEEE Radio Communications, S23–S30 (2005)
2. Briscoe, B., Odlyzko, A., Tilly, B.: Metcalfe's Law is Wrong, IEEE Spectrum Online, [September, 28], Available at HTTP (2006), http://www.spectrum.ieee.org/jul06/4109
3. Bush, S., Kulkarni, A.: Active Networks and Active Network Management: A Proactive Management Framework. Kluwer Academic Publishers, Dordrecht (2001)
4. Estrin, D., Cullar, D., Pister, K., Sukhatme, G.: Connecting the Physical World with Pervasive Networks, Pervasive Computing, pp. 59–69 (2002)
5. Grida, I., Yahia, B., Bertin, E., Crespi, N.: Next/New Generation Networks Services and Management. In: International Conference on Networking and Services ICNS 2006, Silicon Valley, USA (July 16-18, 2006)
6. Gruman, G.: Next challenges for wireless LANs, Recent standards finally answer Wi-Fi security concerns, but as WLANs ramp up and new standards emerge, IT managers need a clear view of the road ahead, InfoWorld, November 26, [September, 21], Available at HTTP (2004) (2006), http://www.infoworld.com/article/04/11/26/48FEwifispecs_1.html?WIRELESS
7. Huang, J.H., Wang, L.-C., Chang, C.J.: Coverage and capacity of a wireless mesh network. 2005 International Conference on Wireless Networks, Communications, and Mobile Computing 1, 458–463 (2005)
8. Hutter, M.: Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability. Springer, Heidelberg (2004)
9. Li, M., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications, 2nd edn. Springer, Heidelberg (1997)
10. Lyonnett, F.: Beyond Classes of Service - Autonomic Networking Management to guarantee the performance of business critical applications in MPLS networks, NGN Summit 2006 Ethernet/IP based Services and Infrastructure, (24 -25 October, 2006 International Congress Centre Munich (ICM) (2006)
11. Naghian, S., Tervonen, J.: Semi-infrastructured mobile ad-hoc mesh networking. In: Proc. IEEE PIMRC 2003, vol. 2, pp. 1069–1073 (2003)
12. The Internet Engineering Task Force, Ad hoc On-Demand Distance Vector (AODV) Routing. [Online], [2006 August, 27], Available at HTTP (July, 2003), http://www.ietf.org/rfc/rfc3561.txt
13. The Internet Engineering Task Force, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR). [Online], [2006 August, 27], Available at HTTP (July, 2004), http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-10.txt
14. The Internet Engineering Task Force, Optimized Link State Routing Protocol (OLSR). [Online], [August 27, 2006], Available at HTTP (October 2003), http://www.ietf.org/rfc/rfc3626.txt
15. Tennenhouse, D.L., et al.: Towards an Active Network Architecture. Computer Communication Review (1996)

16. Tjelta, T.: BROADWAN affordable broadband services for everyone using wireless access networks. In: EUSEA2006, The Euro-Southeast, Asia, Singapore (19th- 23rd June, 2006)
17. Tsai, T.-J., Chen, J.-W.: IEEE 802.11 protocol over wireless mesh networks: problems and perspectives. In: Proc. IEEE AINA 2005, vol. 2, pp. 60–63 (2005)
18. Yahia, B., Bertin, E., Crespi, N.: Towards autonomic management for Next Generation Services. In: International Conference on Networking and Services ICNS 2006, Silicon Valley, USA (July 16-18, 2006)

Fuzzy Integration of Web Data Sources for Data Warehousing

Francisco Araque, Alberto G. Salguero, Ramón Carrasco, and Cecilia Delgado

Dpto. De Lenguajes y Sistemas Informáticos
Universidad de Granada, 18071, Granada (Andalucía), Spain
{faraque, racg, agsh, cdelgado}@ugr.es

Abstract. In this paper we show our work related to an approach for monitoring web sources on the World Wide Web using its temporal properties in order to integrate them in a temporal Data Warehouse. We use these temporal properties obtained for integrating data more efficiently from different data sources according to the requirements of the Data Warehouse administrator. For this integration process we proposed a parallel fuzzy temporal semantic algorithm based on a flexible query language called FSQL in order to obtain more precise data in the Data Warehouse. In addition, we show an application applied to tourism area where data integrated into Data Warehouse can be used to schedule personalized travel as a value-added service for electronic commerce.

Keywords: Data Warehouse, Fuzzy, data sources, data integration, Web.

1 Introduction

Inmon [1] defined a *Data Warehouse* (DW) as "a subject-oriented, integrated, time-variant, non-volatile collection of data in support of management's decision-making process." A DW is a database that stores a copy of operational data with an optimized structure for query and analysis. The scope is one of the issues which define the DW: it is the entire enterprise. The DW is usually implemented using relational databases [2] which define multidimensional structures. Data is extracted from the sources and then loaded into the DW using various data loaders and ETL (Extraction, transformation and loading) tools [3], [4]. After the initial loading, warehouse data must be regularly refreshed, and modifications of operational data since the last DW refreshment must be propagated into the warehouse so that the warehouse data reflects the state of the underlying operational systems [4], [5].

The *World Wide Web* (WWW) has become an important resource of information for Data Warehousing. One drawback to these sources is that they have their own information delivery schedules [6]. Usually, the enterprise develops systems that are continuously polling the sources to enable (near) real-time changes capturing and loading. This approach is not efficient and can produce overload problems. It is more efficient to poll the web site when it is needed. On the other hand, the integration of temporal data from a wide range of WWW data sources is a difficult task and often conventional algorithm discard several refreshments of data sources. This fact causes a less precise DW.

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1208 – 1215, 2007.
© Springer-Verlag Berlin Heidelberg 2007

In order to address the above problems, we propose a system to allow distributed information monitoring of web data sources on the WWW to extract the temporal metadata of them in order to enhance the process of data extraction. Besides, we propose a fuzzy temporal semantic algorithm based on a flexible query language called FSQL in order to obtain more precise data in the DW.

Section 2 introduces a tool for temporal properties extraction. Section 3 describes our proposal for extraction and fuzzy temporal data integration in the DW. Section 4 shows a motivation example. And we finish with the conclusions.

2 Temporal Properties Extraction

In the information extraction process it is necessary to maintain the temporal consistency [4], [7], of the data that the user needs in the sense put forward in real-time applications. In real-time applications [7], the state of the environment as perceived by the controlling system must be consistent with the actual state of the environment being controlled.

To solve these problems, we present tools to define and generate wrappers for Web accessible sources (Web Sources). We define a wrapper interface to specify the capability of Web Sources and extend a wrapper generation toolkit [8]. We use *DETC* (Data Extraction with Temporal Constraints) to refer to the software tool. We can extract some temporal characteristics of the data source, so we can know *when* the data source can offer the data and *how* this data changes along time. This can be represented in the Temporal Component Schema and can be used by the DW administrator (DWA) to decide how scheduling the refreshment activity. It depends on the temporal characteristics of the data source.

In this work, we consider the following temporal parameters (a more detailed explanation can be found in [9]) to be of interest on the basis of the characteristics of the data extraction methods and the data sources:

- *Granularity (Gr)*: It is the extent to which a system contains discrete components of ever-smaller size. In our case, because we are dealing with time, it is common to work with granules like minute, day, month…
- *AW(Availability Window)*. Period of time in which the data source can be accessed by the monitoring programs responsible for data source extraction. There may be more than one daily availability window.
- *M*: time instant when the data source monitoring process is initiated.

By considering the previous temporal properties and two data sources with their specific extraction methods (this can be the same method for both), we can determine the best parameters to integrate data from two sources (according to DWA requirements).

3 Data Integration

Prior to integration, it is necessary to determine under what parameters it is possible to access the sources in search of changes, according to their availability and granularity, obtained automatically by the DECT tool. This process is carried out by the

Temporal Requirements algorithm. It is only possible to determine these parameters previously if there is some pattern related to the source availability.

One of the most complex issues of the integration and transformation interface is the case where there are multiple sources for a single element of data in the DW. For example, in the DW there is a data element that has as its source data element *a1* from legacy application A and a data element *b1* from legacy application B. If it is possible to temporally integrate the data from both sources (on the basis of their temporal properties), semantic integration is undertaken and the result is stored in the DW.

The integration methodology, shown in figure 2, consists of a set of processes that define the rules for capturing a parameter from a single source as well as integrate a set of values semantically equivalent coming from different data sources. It has three phases, shown in figure 2: *Temporal metadata extraction (A), Generation of Refresh meta-data (B) and Refreshment process (C).*

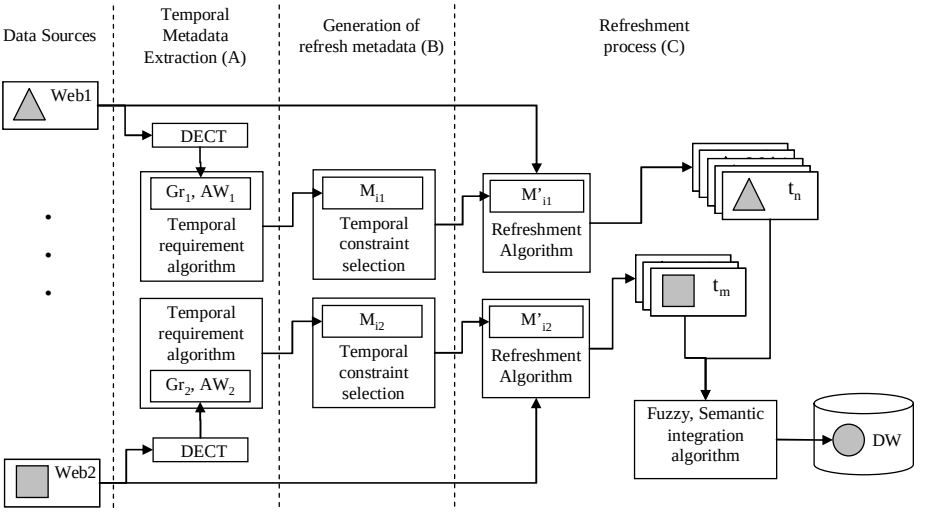


Fig. 2. System architecture

The temporal process of integration can also be divided into three different tasks: the analysis of the temporal requirements, the refreshment of the sources and the semantic integration. The first of the previous tasks verifies that certain temporarl parameters are satisfied, so the refreshment can be carried out. In this phase, the availability as well as other temporal parameters (extraction time, granularity,…) of the source is checked. The result is a set of instants within an interval indicating when the refreshment process should be made. This interval represents the pattern of the source so we only need to study the refreshment process within a finite interval.

In the second phase the most suitable instants are selected to carry out the refreshment process of the data. It is in this second phase where, from the minimum requirements selected by the temporary first stage of integration, the DW designer sets the final refreshment parameters. These parameters can also be set automatically by the system taking care of different criteria (maximum level of detail…).

3.1 Refreshment Process

The third phase is responsible of doing the incremental capture of data sources and to integrate them into a temporal DW table: *id_temporal_DW_table (id_date, fact)*. Where *id_date* is a temporal attribute and *fact* is the integrated data sources for the time *id_date*.

The refreshment algorithm capture the changes of data sources every *Mi* instant for source 1 and *Mj* instant for source 2. Theses sets of instants have been determined in the previous stages. When a change in the data is detected it is stored in a buffer (table). Each data source has an independent buffer so it is possible to have a record in a buffer that should be semantically integrated with many of the other. The Fuzzy Semantic Integration algorithm is responsible of doing this integration.

In this paper, we propose a fuzzy parallel producer consumer scheme to solve this problem. We have used the FSQL language [10] to solve this problem. It extends the SQL language to allow flexible queries. The main extensions are:

- **Linguistic Labels:** They represent a concrete value of the fuzzy attribute. FSQL works with any kind of attributes therefore, by example, a label can have associated: a trapezoidal possibility, a scalar, a text, a XML document, etc.
- **Fuzzy Comparators:** In addition to common comparators (=, >, etc.), FSQL includes fuzzy comparators (Fuzzy equal, Fuzzy greater…). By example FSQL includes fuzzy comparators of two trapezoidal possibility distributions A, B with $A=\$[\alpha_A,\beta_A,\gamma_A,\delta_A]$ $B=\$[\alpha_B,\beta_B,\gamma_B,\delta_B]$.
- **Fulfillment Thresholds γ:** For each simple condition a Fulfillment threshold may be established with the format <condition> THOLD γ, indicating that the condition must be satisfied with a minimum degree γ in [0,1].
- **CDEG(<attribute>) function:** This function shows a column with the Fulfillment degree of the condition of the query for a specific attribute, which is expressed in brackets as the argument.
- **Fuzzy Constants:** We can use and store all of the fuzzy constants: UNKOWN, UNDEFINED, NULL, Fuzzy trapezoid, Linguistic Label, Interval and Approximately values.

The architecture of the Fuzzy Relational Database with the FSQL Server is made up by: data and FSQL Server. The data can be classified in two categories:

Traditional Database: They are data from our relations with a special format to store the fuzzy attribute values. The fuzzy attributes are classified by the system in 4 types:

- *Fuzzy Attributes Type 1*: These attributes are totally crisp, but they have some linguistic trapezoidal labels defined on them, which allow us to make the query conditions for these attributes more flexible.
- *Fuzzy Attributes Type 2*: These attributes admit crisp data as well as possibility distributions over an ordered underlying domain.
- *Fuzzy Attributes Type 3*: These attributes have not an ordered underlying domain. On these attributes, some labels are defined and on these labels, a similarity relation has yet to be defined. With these attributes, we can only use the fuzzy comparator FEQ, as they have no relation of order.

Fuzzy Meta-knowledge Base (FMB)**:** It stores information for the fuzzy treatment of the fuzzy attributes in order to define the:

- *Representation Functions*: These functions are used to show the fuzzy attributes in a comprehensible way for the user and not in the internally used format.
- *Fuzzy Comparison Functions*: They are utilized to compare the fuzzy values and to calculate the compatibility degrees (CDEG function).

At present, we have a FSQL Server available for Oracle© Databases, mainly, programmed in PL/SQL. If there are no errors, the FSQL query is translated into a standard SQL sentence. The resulting SQL sentence includes reference to the Representation and Fuzzy Comparison Functions.

The refreshment algorithm uses two buffers, i.e. two tables id_table_source1 (for source with less granularity time level) and id_table_source_2 (for source with greater granularity time level) with identical structure (id_time_number, data). Where data column is the data value to be integrated and id_time_number is the data time in numerical format. We transform the time in a number using the function DATE_TO_NUMBER. This function returns a number representing the number of days since 1900-Jan-0, plus a fractional portion of a 24 hour day: ddddd.tttttt. This is a crisp attribute but we decide define this as Type 1 in the FMB using the fuzzy constants value #n (approximately n) defined for the DWA depending on the granularity of the data. Also the DWA must to define the fulfillment thresholds γ to this fuzzy comparison to do more o less flexible the time integration. In this definition, is necessary that similarity_times1 value returns by the following SELECT will be always 0:

```
SELECT AVG(CDEG(*)) AS similarity_times1
FROM id_table_source1 a, id_table_source1 b
WHERE a.id_table_source1.id_time_number FEQ
      b.id_time_number THOLD 0
```

Following we explain the Refreshment algorithm which is compose of two parallels threads ($Producer_k$, with k=1,2) for each source and the Fuzzy Semantic Integration algorithm composed of one thread (Consumer):

$Producer_1$

```
For each Mi1 instant
  If a change in the source 1 is detected then
     INSERT INTO id_table_source1  VALUES
     (DATE_TO_NUMBER(time1), data1)

     -- Indicate that there are data available
     Signal continue
```

Producer2

```
For each Mjk instant
  If a change in the source 2 is detected then
     INSERT INTO id_table_source2 VALUES
     (DATE_TO_NUMBER(time2), data2)
```

Consumer

```
Forever
  -- wait until there are data available in source 1
  Wait continue

     -- we obtain the next data of source 1
SELECT id_time_number AS id_time_number1, data AS data1
FROM id_table_source1
    WHERE id_time_number = (SELECT MIN (id_time_number)
    FROM id_table_source1)

    -- wait until all the data of source 2 is available
    to be integrated or discard with data1 of source 1
    Do
       SELECT COUNT(*) AS data_source2_available
       FROM id_table_source2
       WHERE NOT id_time_number FEQ id_time_number1
       THOLD γ AND id_time_number > id_time_number1
    While data_source2_available=0

    -- semantic fuzzy integration into the DW using
    -- an aggregation operator (average…) defined by
    -- the DWA, weighted by the fulfillment degree of
    -- the fuzzy time comparison of both sources
    INSERT INTO id_temporal_DW_table
    SELECT TO_DATE(id_time_number1), AGGREGATION (data,
    data1, CDEG(*))
    FROM id_table_source2 WHERE id_time_number FEQ
    id_time_number1 THOLD γ

    -- delete integrated or discarded data of source 2
    DELETE FROM id_table_source2
    WHERE id_time_number FEQ id_time_number1 THOLD γ OR
    id_time_number < id_time_number1

    -- delete data of source 1
    DELETE FROM id_table_source1
    WHERE id_time_number=id_time_number1
```

4 Example

A Decision Support System (DSS) being based on a DW [11] is presented as an example. This DSS is used in adventure tourism area to assist travel agencies and expert pilots in the decision-making process for a soaring trip [12]. Two web data sources are mainly used to obtain weather information:

- The US National Weather Service Website. We can access weather measurements (temperature, pressure, humidity, etc) in every airport in the world.
- In order to obtain a more detailed data we can access the official city council web.

In order to efficiently integrate these data, it is necessary to use the algorithm described in the previous section.

The DWA introduces the data sources in the *DECT* tool and selects the parameters to integrate. This tool is able to determine the time granularity provided by each data source after a period of time. It uses an algorithm to determine the frequency of the changes produced at the data source. We approximate the granularity of the source by selecting the smallest interval that take place between two consecutive changes.

In the second source, the information about the temperature can be precise with a detail of "minute" (for example, that at 14 hours and 27 minutes there were a temperature of 15ºC), whereas in the first case it talks about the temperature with a detail of "hour" (for example, that at 14 hours there were 15ºC).

Applying the temporal metadata extraction algorithm we would obtain all possible instants for querying both sources. If we employ two minutes on each data source to extract the changes, the algorithm would return "every hour" for the first source (an interval equal to the granularity) and "every two minutes" for the second one. The DWA consider that two minutes is a too high value so in the next phase he or she increase this value to ten minutes. The Refresment algorithm would poll both sources independently according to the previous intervals and filling their respective buffer.

The first source would act as the first "producer" process because it has the lowest granularity level. It has to send a signal to the Fuzzy Semantic Integration algorithm when there is data in the buffers to be integrated (at 15:00, for example). When the signal is received by this algorithm it has to wait until the second source has all of the changes that should be integrated with the last change in the first source.

After that, it selects the last value of the first source and all of the values of the second one which timestamp values are similar, in a certain degree γ, to the timestamp value of the first source. This value is a number between 0 and 1, and has to be specified by the DWA. The DWA has also to set the fuzzy constants value *#n* (approximately *n*, see Table 2) depending on the granularity of the data.

Suppose that, for a margin of #n=0.004 and γ=0.8 values, we have two values in the second source, {1/1/2006-9:59, 17ºC} with a similarity degree value of 0.8 and {1/1/2006-10:05, 18ºC} with a similarity of 0.5, that have to be integrated with the value {1/1/2006-10, 17ºC} of the first source. The DWA has to specify an aggregation function (AVG, MIN…) to integrate these values. The values of the first source can be weighted in the aggregation process according to their similarity degree. If we use the AVG as the aggregation function, the resulting value, stored in the DW for the date 1/1/2006-10, is (*17*+[*17**(0.5/(0.5+0.8))]+[*18**(0.8/(0.5+0.8))]) / 2 = 17.3ºC.

5 Conclusion

In this paper we have presented our work related to an approach for monitoring web sources in order to obtain its temporal properties. We use this information for integrating more efficiently data from different data sources according to the requirements of the DWA. For this integration process we proposed a parallel fuzzy temporal semantic algorithm based on a flexible query language called FSQL in order to obtain more precise data in the DW. The result is more precise because several refreshments of data sources are semantically integrated in a unique DW fact. In a

non-fuzzy algorithm (crisp) several of these refreshments would be discarded. In addition, we show an example applied to tourism area where data integrated into DW can be used to schedule personalized travel as a value-added service for electronic commerce. This work has been supported by the Spanish Research Program under project TIN2005-09098-C05-03 and by Andalucía Research Program under project 2006/282916.

References

1. Inmon, W.H.: Building the Data Warehouse. John Wiley, Chichester (2002)
2. Hammer, J., García-Molina, H., Widom, J., Labio, W., Zhuge, Y.: The Stanford Data Warehousing Project. IEEE Data Engineering Bulletin (1995)
3. Araque, F.: Real-time Data Warehousing with Temporal Requirements. In: Eder, J., Missikoff, M. (eds.) CAiSE 2003. LNCS, vol. 2681, Springer, Heidelberg (2003)
4. Araque, F., Samos, J.: Data warehouse refreshment maintaining temporal consistency. In: 5th Intern. Conference on Enterprise Information Systems, ICEIS 2003, Angers. Fran (2003)
5. Araque, F.: Integrating heterogeneous data sources with temporal constraints using wrappers. In: The 15th Conference On Advanced Information Systems Engineering. Caise Forum. Klagenfurt, Austria (2003)
6. Watanabe, Y., Kitagawa, H., Ishikawa, Y.: Integration of Multiple Dissemination-Based Information Sources Using Source Data Arrival Properties. In: Proc. 2nd Int. Conf. on Web Information Systems Engineering, Kyoto, Japan (2001)
7. Ramamritham, R., Sivasankaran, J.A., Stankovic, D.T., Towsley, M.: Integrating Temporal, Real-Time, and Active Databases. ACM Sigmod Record 25, 8–12 (1996)
8. Sahuguet, A., Azavant, F.: Web Ecology: Recycling HTML pages as XML docu-ments using W4F. In: Informal proceedings of the ACM International Workshop on the Web and Databases (WebDB 1999), Philadelphia, Pennsylvania, USA (1999)
9. Araque, F., Salguero, A.G., Delgado, C., Samos, J.: Algorithms for integrating temporal properties of data in DW. In: 8th Int. Conf. on Enterprise Information Systems (ICEIS). Paphos, Cyprus (May 2006)
10. Carrasco, R.A., Vila, M.A., Galindo, J.: FSQL: a Flexible Query Language for Data Mining. In: Piattini, M., Filipe, J., Braz, J. (eds.) Enterprise Information Systems IV, pp. 68–74. Kluwer Academic Publishers, Dordrecht (2002)
11. March, S.T., Hevner, A.R.: Integrated decision support systems: A data warehousing perspective. Decision Support Systems (2005)
12. Araque, F., Salguero, A., Abad, M.M.: Application of data warehouse and Decision Support System in Soaring site recommendation. In: Proc. Information and Communication Technologies in Tourism, ENTER 2006, Springer, Heidelberg (2006)

Comparison of WiFi Map Construction Methods for WiFi POMDP Navigation Systems

Manuel Ocaña, Luis Miguel Bergasa, Miguel Ángel Sotelo, Ramón Flores, Elena López, and Rafael Barea

Department of Electronics, University of Alcala, Madrid Spain
{mocana,bergasa,sotelo,rflores,elena,barea}@depeca.uah.es

Abstract. The framework of this paper is the robotics navigation inside buildings using WiFi signal strength measure. In most cases this navigation is achieved using a Partially Observable Markov Decission Process (POMDP). In the localization phase of this process the WiFi signal strength is used as observation. The localization system works in two stages: map construction and localization stage. In this paper we compare three different methods for obtaining the WiFi map in the construction stage. The methods have been tested in a real environment using two commercial robotic platforms. Some experimental results and the conclusions are presented.

Keywords: Navigation, WiFi, Localization, POMDP.

1 Introduction

For surveillance robots navigation over huge indoor environments design, in which the objective is to guidance the robot to a goal room using some low level behaviours to perform local navigation, a topological discretization is appropriate to facilitate the planning and learning tasks. A POMDP model provides solutions to localization, planning and learning in this robotic context. These models use probabilistic reasoning process to deal with uncertainties in the actions execution and the observations taken by a robot, very important in the case of WiFi localization sensors.

To find the pose (position and orientation) of a robot from physical sensors is not a trivial problem and is often referred to "the most important problem to provide a mobile robot with autonomous capabilities" [1]. Several systems for localization have been proposed and successfully deployed for an indoor environment. These systems are based on: infrared sensors [2], computer vision [3], ultrasonic sensors [4], laser [5] or radio frequency (RF) [6]. Within the last group we can find localization systems that use WiFi signal strength measure.

WiFi localization systems take advantage of the boom in wireless networks over the last few years. The WiFi networks have become a critical component of the networking infrastructure and are available in most corporate environments (universities, airports, train stations, tribunals, hospitals, etc). Therefore the localization stage can determine the device location without any extra hardware

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1216–1222, 2007.
© Springer-Verlag Berlin Heidelberg 2007

in the environment. It makes these systems attractive for indoor environments where traditional techniques, such as Global Positioning System (GPS) [7] , fail.

In order to estimate the robot location, we propose to measure the WiFi signal strength of received packets in wireless Ethernet interface. This measure depends on the distance and obstacles between wireless Access Points (APs) and the robot. Moreover, the system needs more than one base stations or AP to measure the distance from them to the device. Using these measures they can apply a triangulation algorithm to infer the estimated position [8].

Unfortunately, in indoor environments, the WiFi channel is very noisy and the RF signal can suffer from reflection, diffraction and multipath effect, which makes the signal strength a complex function of distance [6]. To solve this problem, it can be used a priori WiFi map, which represents the signal strength of each AP at certain points in the area of interest [9] [10] [11]. These systems work in two phases: map construction and estimation of the position. During the first phase, the WiFi map is built in a previous setup by mean of different ways. In the estimation phase, the vector of samples received from each access point is compared with the WiFi map and the "nearest" match is returned as the estimated robot location. The problem is that this method involves an enormous calibration effort because the WiFi observations are manually obtained.

In this paper we compare three methods to obtain this WiFi signal strength map. This map will be used in the WiFi POMDP Navigation System. We demonstrate that the automatic training method represents an improvement of the manual and calculated methods and also it manages the adaptability of the map, very important in a WiFi system.

The rest of the paper is organized in the following sections: Section 2 provides a description of the POMDP navigation system. Section 3 explains the WiFi map construction methods comparison and some experimental results, as well as a description of the used test bed. Finally, the conclusions and future work are described in Section 4.

2 WiFi POMDP Navigation System

In this section we provide a resume of our WiFi POMDP Navigation System which was explained by the authors in [12].

When a robot moves across an environment executing several actions (a_t), in execution step t, and it has free of uncertainty in the environment observation, we can modelize this system as a Markov Decision Process (MDP). The MDP is a mathematic model that permit characterize robotics systems without noise in the environment observation. The MDP considers that only the effect of the actions has uncertainty.

When a MDP achieves some execution steps and it goes along a different states $(s_0, s_1 \dots s_n)$ executing some actions $(a_0, a_1 \dots a_n)$, the probability of being in a s_{t+1} state in the $t+1$ execution step is obtained using equation 1.

$$p(s_{t+1}|s_0, a_0, s_1, a_1, \dots, s_t, a_t) = p(s_{t+1}|s_t, a_t) \tag{1}$$

The actions uncertainty model represents the real errors or failures in the execution of the actions. The transition function T incorporates this information to the MDP. In the discrete case, T is a matrix that represents the probability of reaching the state s_{t+1} when the robot is in the state s_t and it has executed the action a_t.

There is a recompense function R for each state s and action a. The robot reaches the maximun value of the recompense function when it reaches the target state travelling through the ideal trajectory and executing the ideal actions.

Although MDP considers that the environment observation is free of uncertainty, in the real robotic systems, there are some uncertainties associated to their sensors observations. These are more significant when the observations are provided by the noisy WiFi sensor [13].

The POMDPs are mathematic models that permit to characterize these noisy systems. A POMDP is defined by the same elements than in a MDP: S (states set), A (actions set), T (transition function), R (recompense function); and then it adds the following elements: O (observations set ($o \in O$)) and ν (observation function).

A POMDP doesn't know its real state because the uncertainty of the observation. A POMDP maintains a belief distribution called $Bel(S)$ or Belief Distribution (Bel) over the states to solve it. This distribution assigns to each state a probability that indicates the possibility of being in the real state. This is the main reason to divide the control stage of a POMDP in two stages, as can be seen in Figure 1:

1. State estimator: the inputs of this block are the current observations and its output is the Bel. This block calculates the probability over all possible states.
2. Politics: the input of this block is the current Bel and its output is the action to perform. This block obtains the optimal action to perform in the next execution step to maximize the recompense (R).

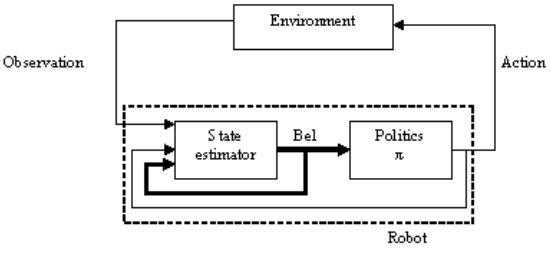


Fig. 1. Partially Observable Markov Decission Process (POMDP)

The state estimator block is known as localization system. This updates the Belief Distribution when a new action or observation is carried out. In the robotics context, these conditions usually are simultaneous. When an action a is

executed and a new observation o is taken, the new probabilities became as it is shown in equation 2.

$$Bel_t(s') = \eta \cdot p(o|s') \cdot \sum_{s \in S} p(s'|s,a) \cdot Bel_{t-1}(s), \forall s' \in S \tag{2}$$

In the context of robot navigation, the states of the Markov model are the localizations (or nodes) of the topological representation of the environment. Actions are local navigations behaviours that the robot executes to move from a state to another, such as move forward (a_F), turn around (a_T), turn to the left (a_L) and turn to the right (a_R). The observations are perceptions of the environment that the robot can extract from its sensors that in our case are obtained from the WiFi ($obs_{WIFI_{AP_x}}$) and Ultrasound (obs_{US}) sensors. In this case, the Markov model is partially observable because the robot never may exactly know the state where the robot is. To solve the POMDP model we have used the WiFi Simultaneous Localization And Mapping (WSLAM) in order to obtain the WiFi observation function and, an extension of the EM algorithm to obtain the Ultrasound observation function.

Observations from the WiFi and the Ultrasound sensors are complementary. The first one obtains an estimation of the global localization and the second one obtains a good estimation of the local environment. The fusion of these observations will produce a good observability of states. POMDP provides a natural way for using multisensorial fusion in their observations models ($p(\vec{o}\,|s)$) by mean of Bayes rule. Assuming that the observations are independent, the observation model can be simplified as in the following way:

$$\begin{aligned} p(\vec{o}\,|s) &= p(obs_{WiFi1}, \ldots, obs_{WiFix}, obs_{US}|s) = \\ &= p(obs_{WiFi1}|s) \cdot \ldots \cdot (obs_{WiFix}|s) \cdot p(obs_{US}|s) \end{aligned} \tag{3}$$

In the next section we show the results of the three methods to obtain a WiFi signal strenght map.

3 Implementation and Results

The Test-Bed environment was established on the 3rd floor of the Polytechnic School building, concretely in the corridor number 4 of the Electronic Department. The layout of this zone is shown in Figure 2. It has a surface of 60m x 60m, with about 50 different rooms, including offices, labs, bathrooms, storerooms and meeting rooms.

Seven Buffalo Access Points (APs) (WBRE-54G) were installed at the all environment. Five APs were connected to omnidirectional antennas and two APs (AP3 and AP7) were connected to antennas of 120 degrees of horizontal beam-width. The APs act as wireless signal transmitters or base stations.

For simplicity, the tests were achieved in the corridor 4. This was discretized into 11 nodes placed at the positions indicated in Figure 2. For each node some radio measures from all the APs in the two main orientations of the corridor

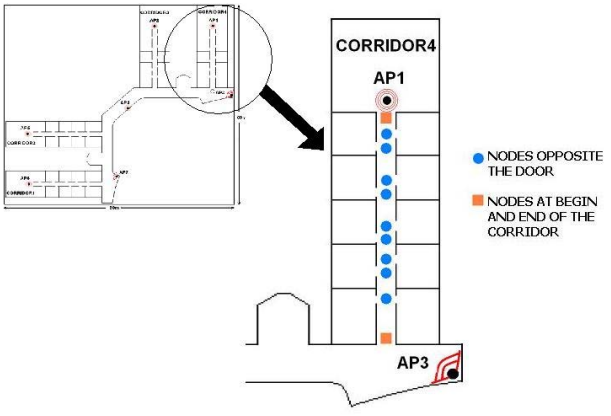


Fig. 2. Test-bed. Department of Electronics (Corridor 4).

were taken to obtain the WiFi radio map, and to extract the results of the WiFi POMDP navigation system.

In order to obtain experimental results of the three methods we used two robots based on the 2AT platform of Activmedia Robotics, as shown in Figure 3. They have the same configuration (Orinoco PCMCIA Gold wireless card, Linux Red Hat 9.0 operating system, wireless tools, a 16 ultrasound sensor ring and a SONY pan-tilt-zoom camera), but in one of them, a metallic structure was added in order to carry out a laptop and to increase the height of the camera.



Fig. 3. Real prototypes used in the results extraction

First, the WiFi map was calculated using a propagation model. The WiFi radio signal propagates through the air following a radio propagation model. This model is very difficult to obtain for indoor environments, due to the multipath suffering and the temporal variation of the WiFi signal. Although an exact and

general model doesn't exist, an approximated model can be used. We calculated the WiFi map with a generic log distance model as shown in equation 4.

$$R_{SL} = T_{SL} + G_{TX} + G_{RX} + 20log(4\pi) - 10n_W logd - X_a \quad (4)$$

Where the R_{SL} is the received signal level, T_{SL} is the transmitted signal level, G_{TX} and G_{RX} are the transmitter and receiver antennas gain respectively, λ is the wavelength (12.5cm for the 2.4GHz of the WiFi signal), n_W is a factor that depends on the walls effect, X_a is a random variable and d is the distance between the emitter and the receiver.

The main advantage of this method is that only required a few seconds of execution and a very slightly work-man effort.

Then, we used the manual training method by mean of positioning the robot along the several states in a manual mode. The robot took 60 WiFi signal samples to calculate the mean value at each state. This needed 9 hours and a half of an intensive man-work.

Finally, we used an automatic training method based on a robust local navigation task to carry out the robot centred along the corridor and using a modified Expectation-Maximization algorithm proposed in a previous work [14]. The user only needed to launch the local navigation application with a slightly supervision during about 2 hours to ensure that the task was carried out correctly by the robot.

The three maps was used in the localization stage of the POMDP for testing the error percentage in this phase. The comparison of these methods is shown in Table 1.

Table 1. Comparison of WiFi Map Construction Methods

Method	Training Time	Man-Work (%)	Error Percentage(%)
Propagation model	$<$ 30 sec	5	98
Manual	9 h 30 min	100	24
Automatic	2 h	10	8

4 Conclusions and Future Works

In this work we have compared three WiFi Map Construction Methods. We have demonstrated that the automatic method reduce the training time of the manual mode. Although the method based on propagation model achieves the best training time, it also achieves the worst error percentage. We conclude that the automatic method is the best compromise between training time, man-work needed and error percentage. In the near future, we have the intention to improve our automatic algorithm to be faster than the current.

Acknowledgment. This work has been funded by grant S-0505/DPI/000176 (Robocity2030 Project) from the Science Department of Community of Madrid, TRA2005-08529-C02-01 (MOVICOM Project) from the Spanish Ministry of Science and Technology (MCyT) and SIMCA (CM-UAH: 2005/018) from the University of Alcala.

References

1. Cox, I.: Blanche-an experiment in guidance and navigation of an autonomous robot vehicle. IEEE Trans. Robot. Automat 7(2), 193–204 (1991)
2. Want, R., Hopper, A., Falco, V., Gibbons, J.: The active badge location system. ACM Transactions on Information Systems 10, 91–102 (1992)
3. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., Shafer, S.: Multi-camera multi-person tracking for easy living. In: Proc. of 3rd IEEE International Workshop on Visual Surveillance, pp. 3–10 (2002)
4. Priyantha, N., Chakraborthy, A., Balakrishnan, H.: The cricket location support system. In: Proc. of the 6th ACM MobiCom, pp. 155–164 (2002)
5. Barber, R., Mata, M., Boada, M., Armingol, J., Salichs, M.: A perception system based on laser information for mobile robot topologic navigation. In: Proc. of 28th Annual Conference of the IEEE Industrial Electronics Society, pp. 2779–2784 (2002)
6. Bahl, P., Padmanabhan, V.: Radar: A, in-building rf-based user location and tracking system. In: Proc. of the IEEE Infocom, pp. 775–784 (2000)
7. Enge, P., Misra, P.: Special issue on gps: The global positioning system. Proc. of the IEEE 87(1), 3–172 (1999)
8. Serrano, O.: Robot localization using wifi signal without intensity map. Proc. of the V Workshop Agentes Físicos (WAF 2004), 79–88 (2004)
9. Howard, A., Siddiqi, S., Sukhatme, G.: An experimental study of localization using wireless ethernet. In: Howard, A., Siddiqi, S., Sukhatme, G. (eds.) Proc. of the International Conference on Field and Service Robotics (July 2003)
10. Ladd, A., Bekris, K., Rudys, A., Marceu, G., Kavraki, L., Wallach, D.: Robotics-based location sensing using wireless ethernet. In: Proc. of the MOBICOM 2002 (2002)
11. Youssef, M., Agrawala, A., Shankar, A.: Wlan location determination via clustering and probability distributions. In: Proc. of the IEEE PerCom 2003, IEEE Computer Society Press, Los Alamitos (2003)
12. Ocaña, M., Bergasa, L., Sotelo, M., Flores, R.: Indoor robot navigation using a pompdp based on wifi and ultrasound observations. In: IROS2005. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 503–509 (August 2005)
13. Ocaña, M., Bergasa, L., Sotelo, M.: Robust navigation indoor using wifi localization. In: MMAR 2004. Proc. of the 10th IEEE Internacional Conference on Methods and Models in Automation and Robotics, pp. 851–856 (August 2004)
14. Ocaña, M., Bergasa, L., Sotelo, M., Flores, R., López, E., Barea, R.: Training method improvements of a wifi navigation system based on POMPDP. In: IROS 2006. Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5259–5264 (October 2006)

A Practical Agent-Based Approach for Pattern Layout Design*

Cunhao Fang[1] and Song Cao[2]

[1] Tsinghua National Laboratory of Information Science and Technology
[2] School of Software, Tsinghua University
100084 Beijing, P.R. China
fangch@tsinghua.edu.cn

Abstract. This paper explores and discusses the application of Software Agent in pattern layout design. First we introduce Pattern Decomposition Representation Model (PDM). By combining Agent technology with the PDM, we also propose and analyze an Agent-Based Pattern Design Model (ABPDM) and present the implementation of the Agents and their cooperation mechanism. Results show that ABPDM is an effective approach for the design of serial and regular pattern.

1 Introduction

In some fields of application in pattern design, such as pattern design of fabric and carpets, there are some principles that are valuable in computer-aided design. First, the elements of pattern, i.e. pattern primitives, are reused frequently. Second, different pattern types contrast sharply and their structures are regular. Thus, the design activities can be formularized clearly. All these would be helpful in automatic and intelligent design.

This paper explores and discusses the application of Software Agent in pattern layout design. First We introduces Pattern Decomposition Representation Model (PDM). Then the Agent-Based Pattern Design Approcah is proposed and the implementation of Agents is presented at the end. We propose an Agent-Based Pattern Design Model (ABPDM), five types of Agents are defined according to the Pattern Decomposition Model (PDM). Each Agent takes charge of specific work automatically. With the cooperation of the Agents, pattern design can be accomplished more quickly. Results show that Agent technology is an effective resolution to the pattern design.

2 Pattern Decomposing Model

By taking the advantages mentioned above, we introduce a model named “Pattern Decomposing Model” (PDM). In this model, the reusable parts of a pattern are

* This paper is sponsored by National Natural Science Foundation of China (NO. 90604027).

R. Moreno-Díaz et al. (Eds.): EUROCAST 2007, LNCS 4739, pp. 1223 – 1228, 2007.
© Springer-Verlag Berlin Heidelberg 2007

extracted as pattern primitives and organized in a Primitive Database. In the meantime, a module, which is separated from specific data, is defined with the abstract structure of the pattern. Therefore, the procedure of pattern design is reduced to two steps: module adjusting and primitive selecting and editing, which enable the changing of the operated object from single line or stroke to the whole primitive. This PDM model also improves the reusability of primitives and reduces the interactive complexity counted in traditional pattern design, so that the speed and quality of pattern generation can be enhanced significantly. PDM is illustrated in **Fig. 1.**

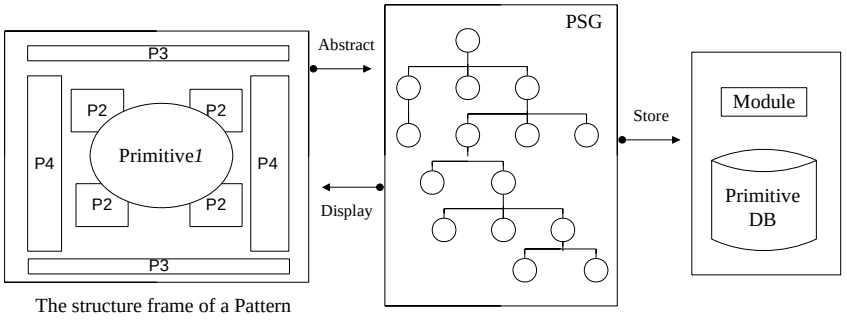


Fig. 1. Pattern Decomposing Model

Based on the model, the pattern can be described with a two-dimensional hierarchical structure, called Pattern Structure Graph (PSG), which usually is acyclic. With this structure, the primitive objects and their relationship can be depicted dynamically. Each node of the PSG represents an object, while the leaf nodes are elementary objects in which the attributes and display information are specified. The non-leaf nodes are managing objects. They indicate the display's layer of the sons and determine the traverse strategy to the descendant. By traversing the son-nodes, the manage object sends the control information to the relative elementary primitive objects, and the receptors trigger some actions according to the current state and the message they get. Therefore, with the PDM model, the interaction between primitive objects and the control mechanism can be implemented practically.

3 Agent-Based Pattern Design Model

Based on the PDM model and Agent theory analyzed above, we construct an Agent-Based Pattern Design Model (ABPDM), in order to support higher reusability, reduce interactive complexity and improve intelligence of design. The architecture of ABPDM is illustrated in **Fig. 2**.

The key component of ABPDM is Agents, which can be divided into five types:

- ***Pattern Primitive Agent (PPA)*** maps the leaf node in PSG and stands for pattern primitives.
- ***Primitive Manage Agent(PMA)*** maps the root in PSG, and is responsible to maintain and harmonize all PPAs.

• ***Interface Agent(IA)*** is responsible to receive the users' input and convert it to PMA to process. IA also serves to PPA's adaption process.

• ***Primitive Search Agent(PSA)*** manages the Primitive Database and automatically selects the primitives according to user's design requirements.

• ***Communication Agent(CA)*** establishes a route and communication between Agents, meanwhile, CA also filters the messages and controls their priorities.

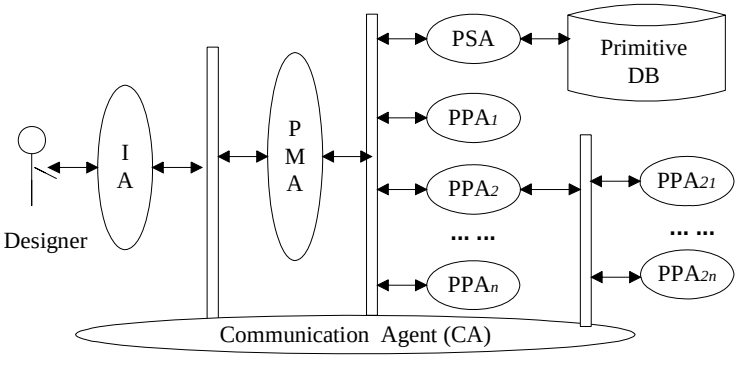


Fig. 2. Architecture of ABPDM

4 Implementation of Agents

4.1 Pattern Primitive Agent: PPA

PPA corresponds to the leaf node in PSG. This kind of Agent only has the representative information of primitives, including some relative computing operations. PPA mainly provides service for the upper Agents.

In pattern design systems, there are three types of primitives used frequently: vector primitives, point-array primitives and literal primitives. According to the type of primitives, PPA manages different primitive data and realizes atomic operations respectively, such as displaying, rotation and translation.

4.2 Primitive Manage Agent: PMA

PMA corresponds to the root in PSG. This kind of Agent is a control Agent, responsible for maintaining and harmonizing all PPAs. By organizing and managing them based on pattern distribution knowledge and the interactive requests submitted by designer, PMA get PPAs to cooperate in order to accomplish some design tasks.

PMA maintains and manages all the PPAs. Its functions fall on the following two aspects:

Task Disassembly: PMA acknowledges the design task received from IA and then disassembles it. With Message Mechanism, the subtasks are submitted for processing.

Coherence Maintenance: PMA monitors all the state changes of PPAs. Based on the internal constraints of Agents, PMA maintains the whole coherence of pattern.

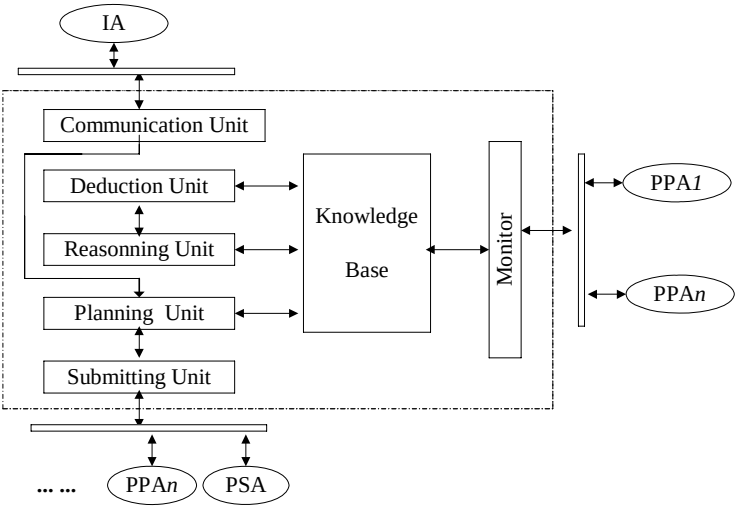


Fig. 3. Implementation of PMA

The workflow of PMA is illustrated in **Fig. 3**.

Communication Unit: It receives the messages from IA.

Deduction Unit: Based on the Knowledge Base, this unit deduces the messages and assembles the design tasks.

Knowledge Base: It stores the requisite knowledge used in deduction, and mainly includes the following:

1) System's Working Status: the statuses of all the registered IA, PPA and CA, such as their position and dimensions.

2) System's Abilities: every Agent's action abilities (operations' prototype) and the history log recording their accomplished tasks.

3) A Set of Predefined Constraints of Pattern Coherence: some constraints of pattern construction, such as the central radiation constraint contained in some pattern; also can be some constraints of field knowledge, for example, the square continuity constraint of pattern in textile industry.

4) Rule Set: It is a set of generation rules. They indicate a series of operations under different prerequisites in order to satisfy the Always constraint. For example:

By defining the internal constraints, the whole coherence of the pattern can be maintained automatically. This method is useful in supplying the gap of messy produced by PDM.

Reasoning Unit: On the basis of the coherent constraints defined in Knowledge Base and the generation rules in Rule Set, this unit maintains the coherence of pattern automatically by reasoning.

Planning Unit: It disassembles the tasks according to the Knowledge Base, and then plans and arranges those subtasks.

Submitting Unit: Using the message mechanism, the unit submits the subtasks to relevant PPAs and PSAs.

Monitor: Snoop on every PPA's state and update the System's Working Status in Knowledge Base in real time.

4.3 Interface Agent: IA

Collaborating with the operating system that manages the human-machine interactive hardware, IA takes the responsibility of receiving the users' inputs and converting them into design tasks. Then, these tasks would be passed on to PMA for processing. IA also serves PPAs and displays the graphs represented in them on the hardware.

4.4 Primitive Search Agent: PSA

PSA manages the Primitive Database and realizes the search strategy accordingly. Based on the user's searching requests, PSA automatically selects the primitives that satisfy the design demands. With the results, PPA is assembled and provided to PMA for pattern distribution. If necessary, PSA would give a candidate set for user to choose.

4.5 Communication Agent: CA

ABPDM does not implement the point-to-point communication between Agents. Instead, all messages are pasted on a blackboard managed by CA. At first, CA establishes a route between Agents, then delivers or broadcasts messages through it. Thus, CA also filters the messages and controls their priorities.

5 Conclusion

In this paper, some research work in the field of applying the Agent theory in the pattern CAD domain is introduced and discussed, which is a new topic. Utilizing the automation and intelligence of Agent, and combining them with the regularity and reusability of pattern design, we propose the Agent-Based Pattern Design Model (ABPDM). In this model, five types of Agents are defined according to the Pattern Decomposing Model (PDM): Pattern Primitive Agent (PPA), Primitive Manage Agent (PMA), Interface Agent (IA), Primitive Search Agent (PSA) and Communication Agent (CA). These Agents construct a hierarchical structure that is similar to PDM's, and each Agent takes charge of some specific work automatically. With the cooperation of the Agents, pattern design is accomplished more quickly, while keeping the quality of opus. Results show that Agent technology is an effective method for pattern design.

References

1. Polzleitner, W., et al.: Invariant pattern location using unsupervised color based perceptual organization and graph2based matching. In: 2001 International Joint Conference on Neural Networks Proceedings, pp. 594–599. IEEE Press, Washington DC (2001)
2. Marculescu, D., Marculescu, R., Khosla, P.K.: Challenges and opportunities in electronic textiles modeling and optimization. In: 39th Proceedings. Design Automation Conference, pp. 175–180 (2002)

3. Rantanen, J., et al.: Smart clothing for the arctic environment. In: The Fourth International Symposium on Wearable Computers Proceedings, pp. 15–23. IEEE Press, Atlanta, GA (2000)
4. Sheng, L., et al.: Application of pattern emulation on weave CAD automatization. In: The 3rd World Congress on Intelligent Control and Automation Proceedings, pp. 2412–2416. IEEE Press, Hefei, China (2000)
5. Cun-hao, F.: The Textile Oriented Pattern Database Design the bachelor degree thesis of Zhejiang University (in Chinese, 1998)
6. Yunhe, P.: Computer Art, Science Press (in Chinese, 1985)

Author Index

Abad-Grau, María M. 187
Abalde, Carlos 217
Abundo, Mario 179
Affenzeller, Michael 747, 788, 812, 820
Agbinya, Johnson 992
Aguado, F. 282
Akama, Kiyoshi 465, 473
Alayón, Francisco 1126
Alba, Enrique 523, 755, 764, 853
Albano, G. 171
Albrecht, Rudolf F. 1
Alemán-Flores, Miguel 960
Alemán-Flores, Patricia 960
Alonso, Javier 408, 1073
Alonso, Jesús B. 740
Alonso, Jos A. 400
Alonso, Miguel A. 257
Álvarez, E. 392
Alvarez, L. 612, 646, 676
Álvarez, Pedro 329
Alvarez, S. 1113
Álvarez–Franco, Fernando J. 845
Álvarez-León, Luis 960
Anzures-García, Mario 57
Aranda, Gonzalo A. 400
Araque, Francisco 415, 1208
Armingol, J.M[a]. 1175
Arroyo, Roberto F. 352
Artho, Cyrille 515

Bañares, José A. 329
Barcala, Fco. Mario 273
Barea, Rafael 684, 1216
Barreira, Noelia 733
Barreiro, Álvaro 225
Barriga, J. 1113
Becker, Bernd 531
Beham, Andreas 747, 829
Bergasa, Luis Miguel 684, 1216
Berkmann, Jens 936, 952
Bernardos, P. 50
Biere, Armin 515
Bischoff, Gabriel P. 505
Blanco Ferro, Antonio 210
Borghoff, Uwe M. 894
Borrego-Díaz, Joaquín 305
Brace, Karl S. 505
Brañas, Enrique Freire 202
Brandl, Bernd 912
Braun, Robin 992, 1199
Braune, Roland 747, 812, 837
Briceño, Juan C. 740
Buchenrieder, Klaus 1057
Buonocore, A. 154

Cabalar, Pedro 241
Caballero-Gil, C. 73
Caballero-Gil, P. 73, 564
Cabodi, Gianpiero 505
Cabrera Gámez, J. 660
Cabrera, J. 1033
Cabrera, Marcelino J. 360
Cabrero Souto, David 202
Canou, Joseph 1066
Cao, Song 481, 1223
Caputo, L. 154
Caraballo, Joaquín 1126
Carrasco, J.P. 1175
Carrasco, Ramón 1208
Carreira, María J. 733
Casanova, José M. 225
Castaño, C.A. 612, 646, 676
Castrillón, M. 1049
Castro, Laura M. 217, 233
Cervantes, A. 755
Češka, Milan 488
Chávez-González, Antonia M. 305
Chaczko, Zenon 105, 992, 1199
Chiang, Frank 992, 1199
Chicano, Francisco 523
Chwatal, Andreas M. 870
Cobos Arribas, Pedro 725
Colombo, Maurizio 448
Crespo, J.L. 50
Cuenca, Carmelo 654, 984

Darriba, Victor M. 297
Davis, David 1191
Delgado, Ana E. 638
Déniz, O. 1049

De Blasio, G. 90
de la Escalera, A. 1175
de La Fortelle, Arnaud 1159
de la Villa, A. 368
de Lope, Javier 138, 621
de Madrid A.P. 65
de Pedro, Teresa 1073
del Castillo, M. Dolores 321
Di Crescenzo, Antonio 163
Dobler, Heinz 130, 376
Domínguez, Eva 273
Dominguez, A. 1033
Domínguez Brito, A.C. 660
Doncel, J.L. 282
Dornaika, Fadi 1105, 1183
Dreiseitl, Stephan 878
Dudek, Roman 654, 984
Dupourqué, Vincent 1066

Ealo, J. 780
Eibensteiner, Florian 1017
Erlebach, Pavel 488
Esclarín Monreal, Julio 668
Esteban-Sánchez, M. Belén 960

Fang, Cunhao 481, 1223
Fernández Alcalá, Mercedes R. 423
Fernández, D. 1113, 1119
Fernández, M. 265
Fernández, Miguel A. 638
Fernández-Caballero, Antonio 638
Fernández de Vega, Francisco 589
Fernández-Parra, Antonio 415
Ferrer, Miguel A. 740
Ferretti, Edgardo 764
Ferro, Antonio Blanco 202
Findenig, Rainer 1017
Flores, Ramón 1119, 1216
Frank, Florian 928
Fredlund, Lars-Åke 539
Freire, J.E. 210
Freire, J.L. 210
Freire-Obregón, D. 90
Fuentes-Pavón, Rafael 960
Fúster-Sabater, A. 564

Galán Moreno, Manuel 1134
Gallardo–Caballero, Ramón 845
Galván, Blas 796
García Antón, Carolina 668
García, Carmelo 1126
García García, Juan Carlos 717
García, M. 612, 646, 676
García, M.A. 1119
García, R. 1119
Garcìa-Cabrera, Lina 384
García–Orellana, Carlos J. 845
García-Rosa, Ricardo 1073
Gavilán, M. 1113, 1119
Gerónimo, David 1105, 1183
Gerla, Václav 195
Gil, Javier Jiménez 1151
Giorno, V. 171
Gómez, J.A. 755
Gómez-Pulido, Juan A. 589, 853
Gómez-Rodríguez, Carlos 257
González Alonso, Ignacio 423
González, Carlos 1073
González, José Luis 360
González-Pachón, Jacinto 42
Gonzílez–Velasco, Horacio M. 845
Graña, J. 290
Granada, Cecilia 1208
Grossmann, Wilfried 345
Gulías, Víctor M. 217, 233, 249
Gutiérrez, Francisco L. 360

Halbach, Mathias 596
Han, Man-Wook 1009
Handzlik, Adam 968
Hein, Werner 936
Hernandez, D. 1033
Herbstritt, Marc 531
Hernández, M. 1049
Hernández, R. 65
Hernández Sosa, J.D. 660
Herrero, Rosa 1105
Hoefer, Gerhard 903
Hoffmann, Rolf 596
Holzer, Andreas 497
Honiden, Shinichi 515
Hornos, Miguel J. 57, 352
Huemer, Mario 928, 936, 944
Humenberger, Martin 1025

Iglesias, Ángel 321
Isasi, P. 755
Isern González, J. 660
Isern, J. 1033

Jablonski, Andrzej 968
Jacak, Witold 772

Jaraíz, M.D. 755
Jiménez Alonso, Felipe 725
Jiménez, A.R. 780
Jiménez, M. Isabel 415
Jodlbauer, Herbert 861

Kinder, Johannes 497
Klempous, Ryszard 629
Kobler, Rene 443
Köckerbauer, Thomas 443
Kopacek, Peter 1009
Kovalerchuk, Boris 26
Krajča, Vladimír 195
Kranzlmüller, Dieter 431
Krissian, K. 612, 646, 676
Kronberger, Gabriel 747, 837
Kubinger, Wilfried 1017, 1025
Küng, Josef 886

Langer, Josef 1017, 1025
León, C. 755
Lhotská, Lenka 195, 337
Lloves, Erica 297
López, Antonio 1105, 1183
López Brugos, José A. 423
López, Elena 684, 1216
López, María T. 638
Lorenzo, J. 1049
Luna, F. 755
Lunglmayr, Michael 952
Luque, C. 755

Macías–Macías, Miguel 845
Mañoso, C. 65
Marín, D. 392
Marrón, M. 1113
Marrón Romera, Marta 717
Martín, A. 368
Martín H., José Antonio 138, 621
Martinelli, Fabio 448
Martinucci, Barbara 163
Martínez-Cantos, J. 701
Martn–Mateos, Francisco J. 400
Mata, Eloy J. 329
Matas-Santiago, Juan M. 589
Matos-Franco, J.C. 804
Mauerkirchner, Manfred 903
Mayr, Herwig 1097
Mayr, Martin 457
Mayrhofer, Rene 920
Mazorra, L. 612, 646, 676
Medina-Medina, Nuria 384
Méndez, Máximo 796
Mehani, Olivier 1159
Miled, Wided 1142
Mira, José 638
Miranda, G. 755
Miró-Julià, Margaret 97
Mocanu, Octavian 1001
Mohylová, Jitka 195
Molina, Guillermo 853
Molina, Juan M. 764
Molina-Gil, J. 73
Molina-Ortiz, Fernando 384
Molinelli, J.M. 282
Molinero, Miguel A. 273
Monteiro, Gonçalo 1089
Montes, Rosana 352
Mora, E. 50
Morales, Antonio 313
Moreno-Díaz, A. 90
Moreno-Díaz, R. 90
Moreno-Díaz jr, Roberto 82
Mori, Paolo 448
Moschner, Markus 345
Moses, Perez 105
Mosshammer, Ralf 928
Munawar, Asim 473
Munetomo, Masaharu 465, 473

Naranjo, A. 1033
Naranjo, José Eugenio 1073, 1113, 1119, 1151
Navarrete, Isabel 313
Nebro, A.J. 755
Neumann, Martin 443
Nikodem, Jan 18, 1167
Nikodem, Maciej 1167
Nishio, Hidenosuke 581
Nistal, José Luis Freire 202
Nobile, A.G. 146
Nunes, Urbano 1089

Ocaña, Manuel 684, 1119, 1216
Oliveira, Luciano 1089
Oliver, Joan 1001
Omasits, Ulrich 443
Otero, J. 290

Paderewski-Rodríguez, Patricia 57
Padrón, Gabino 1126
Parapar, Javier 225

Parent, Michel 1142
Paris, Javier 249
Parra, I. 1113, 1119
Paul, Karel 195
Pazo-Robles, M.E. 564
Peixoto, Paulo 1089
Penas, Marta 733
Penedo, Manuel G. 733
Pérez, G. 282
Pérez, R. 755
Pérez, Ricardo 1126
Pérez de Madrid, A. 65
Pérez-Delgado, M.L. 804
Pérez-Rodríguez, Francisco 589
Pérez Vega, Gilberto 241
Pesquet, Jean Christophe 1142
Petránek, Svojmil 195
Pfaff, Markus 1017
Pichler, Franz 572
Pirozzi, E. 146, 154
Pitzer, Erik 747
Pitzer, Michael 130
Pröll, Karin 772
Priem-Mendes, Silvio 853
Prieto, J.C. 780
Prieto, Laura 337

Quatember, Bernhard 457
Quesada-Arencibia, A. 73
Quesada-Arencibia, Alexis 82
Quintana, Francisca 654, 984

Radi, Harald 920
Raidl, Günther R. 870
Ramler, Rudolf 376
Resconi, Germano 26, 408
Ribadas, Francisco J. 297
Ricciardi, L.M. 146, 154
Rincón, M. 701
Rinnerthaler, Franz 1025
Roa, J.O. 780
Rodríguez-Rodríguez, Jose Carlos 82
Rodríguez-Galiano, María Isabel 42
Rodriguez, G. 368
Romero, M. 65
Rosmanith, Herbert 431
Roszkowska, Elzbieta 1041
Rubio, Julio 329
Rubio Royo, Enrique 1134
Ruipérez, P. 65
Ruiz, Javier Alonso 1151
Ryszard, Klempous 692

Sánchez, J. 612, 646, 676
Sánchez, Javier 709
Sánchez-Gálvez, Luz A. 57
Sánchez-Pérez, Juan M. 589
Sánchez Medina, Javier 1134
Sánchez Penas, Juan José 539
Salgado, Agustín 612, 646, 676, 709
Salguero, Alberto 415
Salguero, Alberto G. 1208
Sallé, Damien 1066
San Martín, Juan José 621
Santana-Montesdeoca, José M. 960
Santiago Jorge, J. 233, 249
Sappa, Angel D. 1105, 1183
Satake, Yuta 465
Saturnino, C. 171
Scharinger, Josef 10
Schleicher, David 684
Schreiner, Wolfgang 443
Sciavicco, Guido 313
Sebastiani, Paola 187
Seco, F. 780
Segura, C. 755
Serrano, J. Ignacio 321
Shibayama, Etsuya 515
Sieniutycz, Stanislaw 34
Simond, Nicolas 1081
Sinz, Carsten 547
Solinis-Camalich, Ignacio 82
Sotelo, M.A. 1113, 1119
Sotelo, Miguel Ángel 1216
Sotelo Vázquez, Miguel Angel 717
Stallinger, H. 944
Stumptner, Reinhard 886
Suárez, José Ignacio 1151
Sulzbachner, Christoph 1025

Tossold, Jürgen 1017
Traonmillin, Marc 1066
Travieso, Carlos M. 740
Triebsees, Thomas 894
Tykierko, Mateusz 113, 121

Uchino, Hiroki 604
Umeo, Hiroshi 604

Vaccarelli, Anna 448
Valderruten, Alberto 249

Vega-Pérez, David 853
Vega-Rodríguez, Miguel A. 589, 853
Veith, Helmut 497
Veselý, Michal 1097
Vidal, C. 282
Vieites, A. 282
Vilares, Manuel 257, 265, 290
Villa Fernández, Benigno 423
Villemonte de la Clergerie, E. 265
Vinagre, Blas 1151
Vinuesa, Javier 1073
Vives, M. Carmen 415
Vojnar, Tomáš 488
Volkert, Jens 431, 443

Wagner, Stefan 747, 788, 812, 820
Weidenhiller, Andreas 861
Wenzel, Dietmar 976
Winkler, Stephan 747, 788, 820
Witschnig, H. 944
Wolfmaier, Klaus 376
Worsch, Thomas 581

Zhang, Wenhui 556
Zhang, Yaoxue 481
Zimmermann, Manfred 936
Zorrilla, M.E. 50, 392
Zweimüller, Boris 515