Lecture Notes in Computer Science 3022

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

## Editorial Board

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*
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
*New York University, NY, 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*

Springer
*Berlin*
*Heidelberg*
*New York*
*Hong Kong*
*London*
*Milan*
*Paris*
*Tokyo*

Tomáš Pajdla  Jiří Matas (Eds.)

# Computer Vision – ECCV 2004

8th European Conference on Computer Vision
Prague, Czech Republic, May 11-14, 2004
Proceedings, Part II

Springer

Volume Editors

Tomáš Pajdla
Jiří Matas
Czech Technical University in Prague, Department of Cybernetics
Center for Machine Perception
121-35 Prague 2, Czech Republic
E-mail: {pajdla,matas}@cmp.felk.cvut.cz

Library of Congress Control Number: 2004104846

CR Subject Classification (1998): I.4, I.3.5, I.5, I.2.9-10

ISSN 0302-9743
ISBN 3-540-21983-8 Springer-Verlag 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-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2004
Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Protago-TeX-Production GmbH
Printed on acid-free paper SPIN: 11007708 06/3142 5 4 3 2 1 0

# Preface

Welcome to the proceedings of the 8th European Conference on Computer Vision!

Following a very successful ECCV 2002, the response to our call for papers was almost equally strong – 555 papers were submitted. We accepted 41 papers for oral and 149 papers for poster presentation.

Several innovations were introduced into the review process. First, the number of program committee members was increased to reduce their review load. We managed to assign to program committee members no more than 12 papers. Second, we adopted a paper ranking system. Program committee members were asked to rank all the papers assigned to them, even those that were reviewed by additional reviewers. Third, we allowed authors to respond to the reviews consolidated in a discussion involving the area chair and the reviewers. Fourth, the reports, the reviews, and the responses were made available to the authors as well as to the program committee members. Our aim was to provide the authors with maximal feedback and to let the program committee members know how authors reacted to their reviews and how their reviews were or were not reflected in the final decision. Finally, we reduced the length of reviewed papers from 15 to 12 pages.

The preparation of ECCV 2004 went smoothly thanks to the efforts of the organizing committee, the area chairs, the program committee, and the reviewers. We are indebted to Anders Heyden, Mads Nielsen, and Henrik J. Nielsen for passing on ECCV traditions and to Dominique Asselineau from ENST/TSI who kindly provided his GestRFIA conference software. We thank Jan-Olof Eklundh and Andrew Zisserman for encouraging us to organize ECCV 2004 in Prague. Andrew Zisserman also contributed many useful ideas concerning the organization of the review process. Olivier Faugeras represented the ECCV Board and helped us with the selection of conference topics. Kyros Kutulakos provided helpful information about the CVPR 2003 organization. David Vernon helped to secure ECVision support.

This conference would never have happened without the support of the Centre for Machine Perception of the Czech Technical University in Prague. We would like to thank Radim Šára for his help with the review process and the proceedings organization. We thank Daniel Večerka and Martin Matoušek who made numerous improvements to the conference software. Petr Pohl helped to put the proceedings together. Martina Budošová helped with administrative tasks. Hynek Bakstein, Ondřej Chum, Jana Kostková, Branislav Mičušík, Štěpán Obdržálek, Jan Šochman, and Vít Zýka helped with the organization.

March 2004 Tomáš Pajdla and Jiří Matas

# Organization

## Conference Chair

| | |
|---|---|
| Václav Hlaváč | CTU Prague, Czech Republic |

## Program Chairs

| | |
|---|---|
| Tomáš Pajdla | CTU Prague, Czech Republic |
| Jiří Matas | CTU Prague, Czech Republic |

## Organization Committee

| | | |
|---|---|---|
| Tomáš Pajdla | | CTU Prague, Czech Republic |
| Radim Šára | Workshops, Tutorials | CTU Prague, Czech Republic |
| Vladimír Smutný | Budget, Exhibition | CTU Prague, Czech Republic |
| Eva Matysková | Local Arrangements | CTU Prague, Czech Republic |
| Jiří Matas | | CTU Prague, Czech Republic |
| Václav Hlaváč | | CTU Prague, Czech Republic |

## Conference Board

| | |
|---|---|
| Hans Burkhardt | University of Freiburg, Germany |
| Bernard Buxton | University College London, UK |
| Roberto Cipolla | University of Cambridge, UK |
| Jan-Olof Eklundh | Royal Institute of Technology, Sweden |
| Olivier Faugeras | INRIA, Sophia Antipolis, France |
| Anders Heyden | Lund University, Sweden |
| Bernd Neumann | University of Hamburg, Germany |
| Mads Nielsen | IT University of Copenhagen, Denmark |
| Giulio Sandini | University of Genoa, Italy |
| David Vernon | Trinity College, Ireland |

## Area Chairs

| | |
|---|---|
| Dmitry Chetverikov | MTA SZTAKI, Hungary |
| Kostas Daniilidis | University of Pennsylvania, USA |
| Rachid Deriche | INRIA Sophia Antipolis, France |
| Jan-Olof Eklundh | KTH Stockholm, Sweden |
| Luc Van Gool | KU Leuven, Belgium & ETH Zürich, Switzerland |
| Richard Hartley | Australian National University, Australia |

| | |
|---|---|
| Michal Irani | Weizmann Institute of Science, Israel |
| Sing Bing Kang | Microsoft Research, USA |
| Aleš Leonardis | University of Ljubljana, Slovenia |
| Stan Li | Microsoft Research China, Beijing, China |
| David Lowe | University of British Columbia, Canada |
| Mads Nielsen | IT University of Copenhagen, Denmark |
| Long Quan | HKUST, Hong Kong, China |
| Jose Santos-Victor | Instituto Superior Tecnico, Portugal |
| Cordelia Schmid | INRIA Rhône-Alpes, France |
| Steven Seitz | University of Washington, USA |
| Amnon Shashua | Hebrew University of Jerusalem, Israel |
| Stefano Soatto | UCLA, Los Angeles, USA |
| Joachim Weickert | Saarland University, Germany |
| Andrew Zisserman | University of Oxford, UK |

## Program Committee

| | | |
|---|---|---|
| Jorgen Ahlberg | Joachim Buhmann | Alexei Efros |
| Narendra Ahuja | Hans Burkhardt | Irfan Essa |
| Yiannis Aloimonos | Aurelio Campilho | Michael Felsberg |
| Arnon Amir | Octavia Camps | Cornelia Fermueller |
| Elli Angelopoulou | Stefan Carlsson | Mario Figueiredo |
| Helder Araujo | Yaron Caspi | Bob Fisher |
| Tal Arbel | Tat-Jen Cham | Andrew Fitzgibbon |
| Karl Astrom | Mike Chantler | David Fleet |
| Shai Avidan | Francois Chaumette | Wolfgang Foerstner |
| Simon Baker | Santanu Choudhury | David Forsyth |
| Subhashis Banerjee | Laurent Cohen | Pascal Fua |
| Kobus Barnard | Michael Cohen | Dariu Gavrila |
| Ronen Basri | Bob Collins | Jan-Mark Geusebroek |
| Serge Belongie | Dorin Comaniciu | Christopher Geyer |
| Marie-Odile Berger | Tim Cootes | Georgy Gimelfarb |
| Horst Bischof | Joao Costeira | Frederic Guichard |
| Michael J. Black | Daniel Cremers | Gregory Hager |
| Andrew Blake | Antonio Criminisi | Allan Hanbury |
| Laure Blanc-Feraud | James Crowley | Edwin Hancock |
| Aaron Bobick | Kristin Dana | Horst Haussecker |
| Rein van den Boomgaard | Trevor Darrell | Eric Hayman |
| Terrance Boult | Larry Davis | Martial Hebert |
| Richard Bowden | Fernando De la Torre | Bernd Heisele |
| Edmond Boyer | Frank Dellaert | Anders Heyden |
| Mike Brooks | Joachim Denzler | Adrian Hilton |
| Michael Brown | Greg Dudek | David Hogg |
| Alfred Bruckstein | Chuck Dyer | Atsushi Imiya |

Michael Isard
Yuri Ivanov
David Jacobs
Allan D. Jepson
Peter Johansen
Nebojsa Jojic
Frederic Jurie
Fredrik Kahl
Daniel Keren
Benjamin Kimia
Ron Kimmel
Nahum Kiryati
Georges Koepfler
Pierre Kornprobst
David Kriegman
Walter Kropatsch
Rakesh Kumar
David Liebowitz
Tony Lindeberg
Jim Little
Yanxi Liu
Yi Ma
Claus Madsen
Tom Malzbender
Jorge Marques
David Marshall
Bogdan Matei
Steve Maybank
Gerard Medioni
Etienne Memin
Rudolf Mester
Krystian Mikolajczyk
J.M.M. Montiel
Theo Moons
Pavel Mrazek
Joe Mundy
Vittorio Murino
David Murray
Hans-Hellmut Nagel
Vic Nalwa
P.J. Narayanan
Nassir Navab
Shree Nayar
Ko Nishino
David Nister
Ole Fogh Olsen
Theodore Papadopoulo
Nikos Paragios
Shmuel Peleg
Francisco Perales
Nicolas Perez
de la Blanca
Pietro Perona
Matti Pietikainen
Filiberto Pla
Robert Pless
Marc Pollefeys
Jean Ponce
Ravi Ramamoorthi
James Rehg
Ian Reid
Tammy Riklin-Raviv
Ehud Rivlin
Nicolas Rougon
Yong Rui
Javier Sanchez
Guillermo Sapiro
Yoichi Sato
Eric Saund
Otmar Scherzer
Bernt Schiele
Mikhail Schlesinger
Christoph Schnoerr
Stan Sclaroff
Mubarak Shah
Eitan Sharon
Jianbo Shi
Kaleem Siddiqi
Cristian Sminchisescu
Nir Sochen
Gerald Sommer
Gunnar Sparr
Jon Sporring
Charles Stewart
Peter Sturm
Changming Sun
Tomas Svoboda
Rahul Swaminathan
Richard Szeliski
Tamas Sziranyi
Chi-keung Tang
Hai Tao
Sibel Tari
Chris Taylor
C.J. Taylor
Bart ter Haar Romeny
Phil Torr
Antonio Torralba
Panos Trahanias
Bill Triggs
Emanuele Trucco
Dimitris Tsakiris
Yanghai Tsin
Matthew Turk
Tinne Tuytelaars
Nuno Vasconcelos
Baba C. Vemuri
David Vernon
Alessandro Verri
Rene Vidal
Jordi Vitria
Yair Weiss
Tomas Werner
Carl-Fredrik Westin
Ross Whitaker
Lior Wolf
Ying Wu
Ming Xie
Ramin Zabih
Assaf Zomet
Steven Zucker

## Additional Reviewers

Lourdes Agapito
Manoj Aggarwal
Parvez Ahammad
Fernando Alegre
Jonathan Alon
Hans Jorgen Andersen
Marco Andreetto
Anelia Angelova
Himanshu Arora
Thangali Ashwin
Vassilis Athitsos
Henry Baird
Harlyn Baker
Evgeniy Bart
Moshe Ben-Ezra
Manuele Bicego
Marten Björkman
Paul Blaer
Ilya Blayvas
Eran Borenstein
Lars Bretzner
Alexia Briassouli
Michael Bronstein
Rupert Brooks
Gabriel Brostow
Thomas Brox
Stephanie Brubaker
Andres Bruhn
Darius Burschka
Umberto Castellani
J.A. Castellanos
James Clark
Andrea Colombari
Marco Cristani
Xiangtian Dai
David Demirdjian
Maxime Descoteaux
Nick Diakopulous
Anthony Dicks
Carlotta Domeniconi
Roman Dovgard
R. Dugad
Ramani Duraiswami
Kerrien Erwan
Claudio Fanti
Michela Farenzena
Doron Feldman
Darya Frolova
Andrea Fusiello
Chunyu Gao
Kshitiz Garg
Yoram Gat
Dan Gelb
Ya'ara Goldschmidt
Michael E. Goss
Leo Grady
Sertan Grigin
Michael Grossberg
J.J. Guerrero
Guodong Guo
Yanlin Guo
Robert Hanek
Matthew Harrison
Tal Hassner
Horst Haussecker
Yakov Hel-Or
Anton van den Hengel
Tat Jen Cham
Peng Chang
John Isidoro
Vishal Jain
Marie-Pierre Jolly
Michael Kaess
Zia Khan
Kristian Kirk
Dan Kong
B. Kröse
Vivek Kwatra
Michael Langer
Catherine Laporte
Scott Larsen
Barbara Levienaise-Obadia
Frederic Leymarie
Fei-Fei Li
Rui Li
Kok-Lim Low
Le Lu
Jocelyn Marchadier
Scott McCloskey
Leonard McMillan
Marci Meingast
Anurag Mittal
Thomas B. Moeslund
Jose Montiel
Philippos Mordohai
Pierre Moreels
Hesam Najafi
P.J. Narayanan
Ara Nefian
Oscar Nestares
Michael Nielsen
Peter Nillius
Fredrik Nyberg
Tom O'Donnell
Eyal Ofek
Takahiro Okabe
Kazunori Okada
D. Ortin
Patrick Perez
Christian Perwass
Carlos Phillips
Srikumar Ramalingam
Alex Rav-Acha
Stefan Roth
Ueli Rutishauser
C. Sagues
Garbis Salgian
Ramin Samadani
Bernard Sarel
Frederik Schaffalitzky
Adam Seeger
Cheng Dong Seon
Ying Shan
Eli Shechtman
Grant Schindler
Nils T. Siebel
Leonid Sigal
Greg Slabaugh
Ben Southall
Eric Spellman
Narasimhan Srinivasa

Drew Steedly
Moritz Stoerring
David Suter
Yi Tan
Donald Tanguay
Matthew Toews
V. Javier Traver
Yaron Ukrainitz
F.E. Wang
Hongcheng Wang
Zhizhou Wang
Joost van de Weijer
Wolfgang Wein
Martin Welk
Michael Werman
Horst Wildenauer
Christopher R. Wren
Ning Xu
Hulya Yalcin
Jingyu Yan
Ruigang Yang
Yll Haxhimusa
Tianli Yu
Lihi Zelnik-Manor
Tao Zhao
Wenyi Zhao
Sean Zhou
Yue Zhou
Ying Zhu

## Sponsors

BIG - Business Information Group a.s.
Camea spol. s r.o.
Casablanca INT s.r.o.
ECVision – European Research Network for Cognitive Computer Vision Systems
Microsoft Research
Miracle Network s.r.o.
Neovision s.r.o.
Toyota

# Table of Contents – Part II

## Geometry

A Generic Concept for Camera Calibration . . . . . 1
*Peter Sturm, Srikumar Ramalingam*

General Linear Cameras . . . . . 14
*Jingyi Yu, Leonard McMillan*

A Framework for Pencil-of-Points Structure-from-Motion . . . . . 28
*Adrien Bartoli, Mathieu Coquerelle, Peter Sturm*

What Do Four Points in Two Calibrated Images Tell Us about the Epipoles? . . . . . 41
*David Nistér, Frederik Schaffalitzky*

## Feature-Based Object Detection and Recognition II

Dynamic Visual Search Using Inner-Scene Similarity: Algorithms and Inherent Limitations . . . . . 58
*Tamar Avraham, Michael Lindenbaum*

Weak Hypotheses and Boosting for Generic Object Detection and Recognition . . . . . 71
*A. Opelt, M. Fussenegger, A. Pinz, P. Auer*

Object Level Grouping for Video Shots . . . . . 85
*Josef Sivic, Frederik Schaffalitzky, Andrew Zisserman*

## Posters II

Statistical Symmetric Shape from Shading for 3D Structure Recovery of Faces . . . . . 99
*Roman Dovgard, Ronen Basri*

Region-Based Segmentation on Evolving Surfaces with Application to 3D Reconstruction of Shape and Piecewise Constant Radiance . . . . . 114
*Hailin Jin, Anthony J. Yezzi, Stefano Soatto*

Human Upper Body Pose Estimation in Static Images . . . . . 126
*Mun Wai Lee, Isaac Cohen*

Automated Optic Disc Localization and Contour Detection Using Ellipse Fitting and Wavelet Transform . . . . . 139
*P.M.D.S. Pallawala, Wynne Hsu, Mong Li Lee, Kah-Guan Au Eong*

View-Invariant Recognition Using Corresponding Object Fragments .... 152
*Evgeniy Bart, Evgeny Byvatov, Shimon Ullman*

Variational Pairing of Image Segmentation and Blind Restoration ..... 166
*Leah Bar, Nir Sochen, Nahum Kiryati*

Towards Intelligent Mission Profiles of Micro Air Vehicles:
Multiscale Viterbi Classification ..... 178
*Sinisa Todorovic, Michael C. Nechyba*

Stitching and Reconstruction of Linear-Pushbroom Panoramic Images
for Planar Scenes ..... 190
*Chu-Song Chen, Yu-Ting Chen, Fay Huang*

Audio-Video Integration for Background Modelling ..... 202
*Marco Cristani, Manuele Bicego, Vittorio Murino*

A Combined PDE and Texture Synthesis Approach to Inpainting ..... 214
*Harald Grossauer*

Face Recognition from Facial Surface Metric ..... 225
*Alexander M. Bronstein, Michael M. Bronstein, Alon Spira,
Ron Kimmel*

Image and Video Segmentation by Anisotropic Kernel Mean Shift ..... 238
*Jue Wang, Bo Thiesson, Yingqing Xu, Michael Cohen*

Colour Texture Segmentation by Region-Boundary Cooperation ..... 250
*Jordi Freixenet, Xavier Muñoz, Joan Martí, Xavier Lladó*

Spectral Solution of Large-Scale Extrinsic Camera Calibration as
a Graph Embedding Problem ..... 262
*Matthew Brand, Matthew Antone, Seth Teller*

Estimating Intrinsic Images from Image Sequences with
Biased Illumination ..... 274
*Yasuyuki Matsushita, Stephen Lin, Sing Bing Kang, Heung-Yeung Shum*

Structure and Motion from Images of Smooth Textureless Objects ..... 287
*Yasutaka Furukawa, Amit Sethi, Jean Ponce, David Kriegman*

Automatic Non-rigid 3D Modeling from Video ..... 299
*Lorenzo Torresani, Aaron Hertzmann*

From a 2D Shape to a String Structure Using the Symmetry Set ..... 313
*Arjan Kuijper, Ole Fogh Olsen, Peter Giblin, Philip Bille,
Mads Nielsen*

Extrinsic Camera Parameter Recovery from Multiple Image Sequences Captured by an Omni-directional Multi-camera System ..... 326
*Tomokazu Sato, Sei Ikeda, Naokazu Yokoya*

Evaluation of Robust Fitting Based Detection ..... 341
*Sio-Song Ieng, Jean-Philippe Tarel, Pierre Charbonnier*

Local Orientation Smoothness Prior for Vascular Segmentation of Angiography ..... 353
*Wilbur C.K. Wong, Albert C.S. Chung, Simon C.H. Yu*

Weighted Minimal Hypersurfaces and Their Applications in Computer Vision ..... 366
*Bastian Goldlücke, Marcus Magnor*

Interpolating Novel Views from Image Sequences by Probabilistic Depth Carving ..... 379
*Annie Yao, Andrew Calway*

Sparse Finite Elements for Geodesic Contours with Level-Sets ..... 391
*Martin Weber, Andrew Blake, Roberto Cipolla*

Hierarchical Implicit Surface Joint Limits to Constrain Video-Based Motion Capture ..... 405
*Lorna Herda, Raquel Urtasun, Pascal Fua*

Separating Specular, Diffuse, and Subsurface Scattering Reflectances from Photometric Images ..... 419
*Tai-Pang Wu, Chi-Keung Tang*

Temporal Factorization vs. Spatial Factorization ..... 434
*Lihi Zelnik-Manor, Michal Irani*

Tracking Aspects of the Foreground against the Background ..... 446
*Hieu T. Nguyen, Arnold Smeulders*

Example-Based Stereo with General BRDFs ..... 457
*Adrien Treuille, Aaron Hertzmann, Steven M. Seitz*

Adaptive Probabilistic Visual Tracking with Incremental Subspace Update ..... 470
*David Ross, Jongwoo Lim, Ming-Hsuan Yang*

On Refractive Optical Flow ..... 483
*Sameer Agarwal, Satya P. Mallick, David Kriegman, Serge Belongie*

Matching Tensors for Automatic Correspondence and Registration ..... 495
*Ajmal S. Mian, Mohammed Bennamoun, Robyn Owens*

A Biologically Motivated and Computationally Tractable Model of Low and Mid-Level Vision Tasks . . . . . 506
*Iasonas Kokkinos, Rachid Deriche, Petros Maragos, Olivier Faugeras*

Appearance Based Qualitative Image Description for Object Class Recognition . . . . . 518
*Johan Thureson, Stefan Carlsson*

Consistency Conditions on the Medial Axis . . . . . 530
*Anthony Pollitt, Peter Giblin, Benjamin Kimia*

Normalized Cross-Correlation for Spherical Images . . . . . 542
*Lorenzo Sorgi, Kostas Daniilidis*

Bias in the Localization of Curved Edges . . . . . 554
*Paulo R.S. Mendonça, Dirk Padfield, James Miller, Matt Turek*

## Texture

Texture Boundary Detection for Real-Time Tracking . . . . . 566
*Ali Shahrokni, Tom Drummond, Pascal Fua*

A TV Flow Based Local Scale Measure for Texture Discrimination . . . . . 578
*Thomas Brox, Joachim Weickert*

Spatially Homogeneous Dynamic Textures . . . . . 591
*Gianfranco Doretto, Eagle Jones, Stefano Soatto*

Synthesizing Dynamic Texture with Closed-Loop Linear Dynamic System . . . . . 603
*Lu Yuan, Fang Wen, Ce Liu, Heung-Yeung Shum*

**Author Index** . . . . . 617

# Table of Contents – Part I

## Tracking I

A Unified Algebraic Approach to 2-D and 3-D Motion Segmentation .... 1
*René Vidal, Yi Ma*

Enhancing Particle Filters Using Local Likelihood Sampling ..... 16
*Péter Torma, Csaba Szepesvári*

A Boosted Particle Filter: Multitarget Detection and Tracking ..... 28
*Kenji Okuma, Ali Taleghani, Nando de Freitas, James J. Little, David G. Lowe*

## Feature-Based Object Detection and Recognition I

Simultaneous Object Recognition and Segmentation by Image Exploration ..... 40
*Vittorio Ferrari, Tinne Tuytelaars, Luc Van Gool*

Recognition by Probabilistic Hypothesis Construction ..... 55
*Pierre Moreels, Michael Maire, Pietro Perona*

Human Detection Based on a Probabilistic Assembly of Robust Part Detectors ..... 69
*Krystian Mikolajczyk, Cordelia Schmid, Andrew Zisserman*

## Posters I

Model Selection for Range Segmentation of Curved Objects ..... 83
*Alireza Bab-Hadiashar, Niloofar Gheissari*

High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging ..... 95
*Changsoo Je, Sang Wook Lee, Rae-Hong Park*

Using Inter-feature-Line Consistencies for Sequence-Based Object Recognition ..... 108
*Jiun-Hung Chen, Chu-Song Chen*

Discriminant Analysis on Embedded Manifold ..... 121
*Shuicheng Yan, Hongjiang Zhang, Yuxiao Hu, Benyu Zhang, Qiansheng Cheng*

Multiscale Inverse Compositional Alignment for Subdivision Surface Maps . . . . . 133
*Igor Guskov*

A Fourier Theory for Cast Shadows . . . . . 146
*Ravi Ramamoorthi, Melissa Koudelka, Peter Belhumeur*

Surface Reconstruction by Propagating 3D Stereo Data in Multiple 2D Images . . . . . 163
*Gang Zeng, Sylvain Paris, Long Quan, Maxime Lhuillier*

Visibility Analysis and Sensor Planning in Dynamic Environments . . . . . 175
*Anurag Mittal, Larry S. Davis*

Camera Calibration from the Quasi-affine Invariance of Two Parallel Circles . . . . . 190
*Yihong Wu, Haijiang Zhu, Zhanyi Hu, Fuchao Wu*

Texton Correlation for Recognition . . . . . 203
*Thomas Leung*

Multiple View Feature Descriptors from Image Sequences via Kernel Principal Component Analysis . . . . . 215
*Jason Meltzer, Ming-Hsuan Yang, Rakesh Gupta, Stefano Soatto*

An Affine Invariant Salient Region Detector . . . . . 228
*Timor Kadir, Andrew Zisserman, Michael Brady*

A Visual Category Filter for Google Images . . . . . 242
*Robert Fergus, Pietro Perona, Andrew Zisserman*

Scene and Motion Reconstruction from Defocused and Motion-Blurred Images via Anisotropic Diffusion . . . . . 257
*Paolo Favaro, Martin Burger, Stefano Soatto*

Semantics Discovery for Image Indexing . . . . . 270
*Joo-Hwee Lim, Jesse S. Jin*

Hand Gesture Recognition within a Linguistics-Based Framework . . . . . 282
*Konstantinos G. Derpanis, Richard P. Wildes, John K. Tsotsos*

Line Geometry for 3D Shape Understanding and Reconstruction . . . . . 297
*Helmut Pottmann, Michael Hofer, Boris Odehnal, Johannes Wallner*

Extending Interrupted Feature Point Tracking for 3-D Affine Reconstruction . . . . . 310
*Yasuyuki Sugaya, Kenichi Kanatani*

Many-to-Many Feature Matching Using Spherical Coding of Directed Graphs . . . . . 322
*M. Fatih Demirci, Ali Shokoufandeh, Sven Dickinson, Yakov Keselman, Lars Bretzner*

Coupled-Contour Tracking through Non-orthogonal Projections and Fusion for Echocardiography . . . . . 336
*Xiang Sean Zhou, Dorin Comaniciu, Sriram Krishnan*

A Statistical Model for General Contextual Object Recognition . . . . . 350
*Peter Carbonetto, Nando de Freitas, Kobus Barnard*

Reconstruction from Projections Using Grassmann Tensors . . . . . 363
*Richard I. Hartley, Fred Schaffalitzky*

Co-operative Multi-target Tracking and Classification . . . . . 376
*Pankaj Kumar, Surendra Ranganath, Kuntal Sengupta, Huang Weimin*

A Linguistic Feature Vector for the Visual Interpretation of Sign Language . . . . . 390
*Richard Bowden, David Windridge, Timor Kadir, Andrew Zisserman, Michael Brady*

Fast Object Detection with Occlusions . . . . . 402
*Yen-Yu Lin, Tyng-Luh Liu, Chiou-Shann Fuh*

Pose Estimation of Free-Form Objects . . . . . 414
*Bodo Rosenhahn, Gerald Sommer*

Interactive Image Segmentation Using an Adaptive GMMRF Model . . . . . 428
*Andrew Blake, Carsten Rother, M. Brown, Patrick Perez, Philip Torr*

Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model . . . . . 442
*Xianghua Ying, Zhanyi Hu*

Image Clustering with Metric, Local Linear Structure, and Affine Symmetry . . . . . 456
*Jongwoo Lim, Jeffrey Ho, Ming-Hsuan Yang, Kuang-chih Lee, David Kriegman*

Face Recognition with Local Binary Patterns . . . . . 469
*Timo Ahonen, Abdenour Hadid, Matti Pietikäinen*

Steering in Scale Space to Optimally Detect Image Structures . . . . . 482
*Jeffrey Ng, Anil A. Bharath*

Hand Motion from 3D Point Trajectories and a Smooth Surface Model . . . . . 495
*Guillaume Dewaele, Frédéric Devernay, Radu Horaud*

A Robust Probabilistic Estimation Framework for Parametric Image Models ..... 508
*Maneesh Singh, Himanshu Arora, Narendra Ahuja*

Keyframe Selection for Camera Motion and Structure Estimation from Multiple Views ..... 523
*Thorsten Thormählen, Hellward Broszio, Axel Weissenfeld*

Omnidirectional Vision: Unified Model Using Conformal Geometry ..... 536
*Eduardo Bayro-Corrochano, Carlos López-Franco*

A Robust Algorithm for Characterizing Anisotropic Local Structures .... 549
*Kazunori Okada, Dorin Comaniciu, Navneet Dalal, Arun Krishnan*

Dimensionality Reduction by Canonical Contextual Correlation Projections ..... 562
*Marco Loog, Bram van Ginneken, Robert P.W. Duin*

**Illumination, Reflectance, and Reflection**

Accuracy of Spherical Harmonic Approximations for Images of Lambertian Objects under Far and Near Lighting ..... 574
*Darya Frolova, Denis Simakov, Ronen Basri*

Characterization of Human Faces under Illumination Variations Using Rank, Integrability, and Symmetry Constraints..... 588
*S. Kevin Zhou, Rama Chellappa, David W. Jacobs*

User Assisted Separation of Reflections from a Single Image Using a Sparsity Prior..... 602
*Anat Levin, Yair Weiss*

The Quality of Catadioptric Imaging – Application to Omnidirectional Stereo ..... 614
*Wolfgang Stürzl, Hansjürgen Dahmen, Hanspeter A. Mallot*

**Author Index** ..... 629

# Table of Contents – Part III

## Learning and Recognition

A Constrained Semi-supervised Learning Approach to Data Association . . . . . 1
*Hendrik Kück, Peter Carbonetto, Nando de Freitas*

Learning Mixtures of Weighted Tree-Unions by Minimizing Description Length . . . . . 13
*Andrea Torsello, Edwin R. Hancock*

Decision Theoretic Modeling of Human Facial Displays . . . . . 26
*Jesse Hoey, James J. Little*

Kernel Feature Selection with Side Data Using a Spectral Approach . . . . . 39
*Amnon Shashua, Lior Wolf*

## Tracking II

Tracking Articulated Motion Using a Mixture of Autoregressive Models . . . . . 54
*Ankur Agarwal, Bill Triggs*

Novel Skeletal Representation for Articulated Creatures . . . . . 66
*Gabriel J. Brostow, Irfan Essa, Drew Steedly, Vivek Kwatra*

An Accuracy Certified Augmented Reality System for Therapy Guidance . . . . . 79
*Stéphane Nicolau, Xavier Pennec, Luc Soler, Nichlas Ayache*

## Posters III

3D Human Body Tracking Using Deterministic Temporal Motion Models . . . . . 92
*Raquel Urtasun, Pascal Fua*

Robust Fitting by Adaptive-Scale Residual Consensus . . . . . 107
*Hanzi Wang, David Suter*

Causal Camera Motion Estimation by Condensation and Robust Statistics Distance Measures . . . . . 119
*Tal Nir, Alfred M. Bruckstein*

An Adaptive Window Approach for Image Smoothing and Structures Preserving . . . . . 132
*Charles Kervrann*

Extraction of Semantic Dynamic Content from Videos with Probabilistic Motion Models . . . . . 145
*Gwenaëlle Piriou, Patrick Bouthemy, Jian-Feng Yao*

Are Iterations and Curvature Useful for Tensor Voting? . . . . . 158
*Sylvain Fischer, Pierre Bayerl, Heiko Neumann, Gabriel Cristóbal, Rafael Redondo*

A Feature-Based Approach for Determining Dense Long Range Correspondences . . . . . 170
*Josh Wills, Serge Belongie*

Combining Geometric- and View-Based Approaches for Articulated Pose Estimation . . . . . 183
*David Demirdjian*

Shape Matching and Recognition – Using Generative Models and Informative Features . . . . . 195
*Zhuowen Tu, Alan L. Yuille*

Generalized Histogram: Empirical Optimization of Low Dimensional Features for Image Matching . . . . . 210
*Shin'ichi Satoh*

Recognizing Objects in Range Data Using Regional Point Descriptors . . . 224
*Andrea Frome, Daniel Huber, Ravi Kolluri, Thomas Bülow, Jitendra Malik*

Shape Reconstruction from 3D and 2D Data Using PDE-Based Deformable Surfaces . . . . . 238
*Ye Duan, Liu Yang, Hong Qin, Dimitris Samaras*

Structure and Motion Problems for Multiple Rigidly Moving Cameras . . . 252
*Henrik Stewenius, Kalle Åström*

Detection and Tracking Scheme for Line Scratch Removal in an Image Sequence . . . . . 264
*Bernard Besserer, Cedric Thiré*

Color Constancy Using Local Color Shifts . . . . . 276
*Marc Ebner*

Image Anisotropic Diffusion Based on Gradient Vector Flow Fields . . . . . 288
*Hongchuan Yu, Chin-Seng Chua*

Optimal Importance Sampling for Tracking in Image Sequences: Application to Point Tracking ..... 302
*Elise Arnaud, Etienne Mémin*

Learning to Segment ..... 315
*Eran Borenstein, Shimon Ullman*

MCMC-Based Multiview Reconstruction of Piecewise Smooth Subdivision Curves with a Variable Number of Control Points ..... 329
*Michael Kaess, Rafal Zboinski, Frank Dellaert*

Bayesian Correction of Image Intensity with Spatial Consideration ..... 342
*Jiaya Jia, Jian Sun, Chi-Keung Tang, Heung-Yeung Shum*

Stretching Bayesian Learning in the Relevance Feedback of Image Retrieval ..... 355
*Ruofei Zhang, Zhongfei (Mark) Zhang*

Real-Time Tracking of Multiple Skin-Colored Objects with a Possibly Moving Camera ..... 368
*Antonis A. Argyros, Manolis I.A. Lourakis*

Evaluation of Image Fusion Performance with Visible Differences ..... 380
*Vladimir Petrović, Costas Xydeas*

An Information-Based Measure for Grouping Quality ..... 392
*Erik A. Engbers, Michael Lindenbaum, Arnold W.M. Smeulders*

Bias in Shape Estimation ..... 405
*Hui Ji, Cornelia Fermüller*

Contrast Marginalised Gradient Template Matching ..... 417
*Saleh Basalamah, Anil Bharath, Donald McRobbie*

The Kullback-Leibler Kernel as a Framework for Discriminant and Localized Representations for Visual Recognition ..... 430
*Nuno Vasconcelos, Purdy Ho, Pedro Moreno*

Partial Object Matching with Shapeme Histograms ..... 442
*Y. Shan, H.S. Sawhney, B. Matei, R. Kumar*

Modeling and Synthesis of Facial Motion Driven by Speech ..... 456
*Payam Saisan, Alessandro Bissacco, Alessandro Chiuso, Stefano Soatto*

Recovering Local Shape of a Mirror Surface from Reflection of a Regular Grid ..... 468
*Silvio Savarese, Min Chen, Pietro Perona*

Structure of Applicable Surfaces from Single Views . . . . . 482
*Nail Gumerov, Ali Zandifar, Ramani Duraiswami, Larry S. Davis*

Joint Bayes Filter: A Hybrid Tracker for Non-rigid Hand Motion Recognition . . . . . 497
*Huang Fei, Ian Reid*

Iso-disparity Surfaces for General Stereo Configurations . . . . . 509
*Marc Pollefeys, Sudipta Sinha*

Camera Calibration with Two Arbitrary Coplanar Circles . . . . . 521
*Qian Chen, Haiyuan Wu, Toshikazu Wada*

Reconstruction of 3-D Symmetric Curves from Perspective Images without Discrete Features . . . . . 533
*Wei Hong, Yi Ma, Yizhou Yu*

A Topology Preserving Non-rigid Registration Method Using a Symmetric Similarity Function-Application to 3-D Brain Images . . . . . 546
*Vincent Noblet, Christian Heinrich, Fabrice Heitz, Jean-Paul Armspach*

A Correlation-Based Approach to Robust Point Set Registration . . . . . 558
*Yanghai Tsin, Takeo Kanade*

Hierarchical Organization of Shapes for Efficient Retrieval . . . . . 570
*Shantanu Joshi, Anuj Srivastava, Washington Mio, Xiuwen Liu*

**Information-Based Image Processing**

Intrinsic Images by Entropy Minimization . . . . . 582
*Graham D. Finlayson, Mark S. Drew, Cheng Lu*

Image Similarity Using Mutual Information of Regions . . . . . 596
*Daniel B. Russakoff, Carlo Tomasi, Torsten Rohlfing, Calvin R. Maurer, Jr.*

**Author Index** . . . . . 609

# Table of Contents – Part IV

## Scale Space, Flow, Restoration

A $l^1$-Unified Variational Framework for Image Restoration ..... 1
*Julien Bect, Laure Blanc-Féraud, Gilles Aubert, Antonin Chambolle*

Support Blob Machines. The Sparsification of Linear Scale Space ..... 14
*Marco Loog*

High Accuracy Optical Flow Estimation Based on a Theory for Warping ..... 25
*Thomas Brox, Andrés Bruhn, Nils Papenberg, Joachim Weickert*

Model-Based Approach to Tomographic Reconstruction Including Projection Deblurring. Sensitivity of Parameter Model to Noise on Data ..... 37
*Jean Michel Lagrange, Isabelle Abraham*

## 2D Shape Detection and Recognition

Unlevel-Sets: Geometry and Prior-Based Segmentation ..... 50
*Tammy Riklin-Raviv, Nahum Kiryati, Nir Sochen*

Learning and Bayesian Shape Extraction for Object Recognition ..... 62
*Washington Mio, Anuj Srivastava, Xiuwen Liu*

Multiphase Dynamic Labeling for Variational Recognition-Driven Image Segmentation ..... 74
*Daniel Cremers, Nir Sochen, Christoph Schnörr*

## Posters IV

Integral Invariant Signatures ..... 87
*Siddharth Manay, Byung-Woo Hong, Anthony J. Yezzi, Stefano Soatto*

Detecting Keypoints with Stable Position, Orientation, and Scale under Illumination Changes ..... 100
*Bill Triggs*

Spectral Simplification of Graphs ..... 114
*Huaijun Qiu, Edwin R. Hancock*

Inferring White Matter Geometry from Diffusion Tensor MRI: Application to Connectivity Mapping ..... 127
*Christophe Lenglet, Rachid Deriche, Olivier Faugeras*

Unifying Approaches and Removing Unrealistic Assumptions in Shape from Shading: Mathematics Can Help . . . . . 141
*Emmanuel Prados, Olivier Faugeras*

Morphological Operations on Matrix-Valued Images . . . . . 155
*Bernhard Burgeth, Martin Welk, Christian Feddern, Joachim Weickert*

Constraints on Coplanar Moving Points . . . . . 168
*Sujit Kuthirummal, C.V. Jawahar, P.J. Narayanan*

A PDE Solution of Brownian Warping . . . . . 180
*Mads Nielsen, P. Johansen*

Stereovision-Based Head Tracking Using Color and Ellipse Fitting in a Particle Filter . . . . . 192
*Bogdan Kwolek*

Parallel Variational Motion Estimation by Domain Decomposition and Cluster Computing . . . . . 205
*Timo Kohlberger, Christoph Schnörr, Andrés Bruhn, Joachim Weickert*

Whitening for Photometric Comparison of Smooth Surfaces under Varying Illumination . . . . . 217
*Margarita Osadchy, Michael Lindenbaum, David Jacobs*

Structure from Motion of Parallel Lines . . . . . 229
*Patrick Baker, Yiannis Aloimonos*

A Bayesian Framework for Multi-cue 3D Object Tracking . . . . . 241
*Jan Giebel, Darin M. Gavrila, Christoph Schnörr*

On the Significance of Real-World Conditions for Material Classification . . . . . 253
*Eric Hayman, Barbara Caputo, Mario Fritz, Jan-Olof Eklundh*

Toward Accurate Segmentation of the LV Myocardium and Chamber for Volumes Estimation in Gated SPECT Sequences . . . . . 267
*Diane Lingrand, Arnaud Charnoz, Pierre Malick Koulibaly, Jacques Darcourt, Johan Montagnat*

An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets . . . . . 279
*Zia Khan, Tucker Balch, Frank Dellaert*

Human Pose Estimation Using Learnt Probabilistic Region Similarities and Partial Configurations . . . . . 291
*Timothy J. Roberts, Stephen J. McKenna, Ian W. Ricketts*

Tensor Field Segmentation Using Region Based Active Contour Model . . . . . 304
*Zhizhou Wang, Baba C. Vemuri*

Groupwise Diffeomorphic Non-rigid Registration for Automatic Model Building . . . . . 316
*T.F. Cootes, S. Marsland, C.J. Twining, K. Smith, C.J. Taylor*

Separating Transparent Layers through Layer Information Exchange . . . . 328
*Bernard Sarel, Michal Irani*

Multiple Classifier System Approach to Model Pruning in Object Recognition . . . . . 342
*Josef Kittler, Ali R. Ahmadyfard*

Coaxial Omnidirectional Stereopsis . . . . . 354
*Libor Spacek*

Classifying Materials from Their Reflectance Properties . . . . . 366
*Peter Nillius, Jan-Olof Eklundh*

Seamless Image Stitching in the Gradient Domain . . . . . 377
*Anat Levin, Assaf Zomet, Shmuel Peleg, Yair Weiss*

Spectral Clustering for Robust Motion Segmentation . . . . . 390
*JinHyeong Park, Hongyuan Zha, Rangachar Kasturi*

Learning Outdoor Color Classification from Just One Training Image . . . . 402
*Roberto Manduchi*

A Polynomial-Time Metric for Attributed Trees . . . . . 414
*Andrea Torsello, Džena Hidović, Marcello Pelillo*

Probabilistic Multi-view Correspondence in a Distributed Setting with No Central Server . . . . . 428
*Shai Avidan, Yael Moses, Yoram Moses*

Monocular 3D Reconstruction of Human Motion in Long Action Sequences . . . . . 442
*Gareth Loy, Martin Eriksson, Josephine Sullivan, Stefan Carlsson*

Fusion of Infrared and Visible Images for Face Recognition . . . . . 456
*Aglika Gyaourova, George Bebis, Ioannis Pavlidis*

Reliable Fiducial Detection in Natural Scenes . . . . . 469
*David Claus, Andrew W. Fitzgibbon*

Light Field Appearance Manifolds . . . . . 481
*Chris Mario Christoudias, Louis-Philippe Morency, Trevor Darrell*

Galilean Differential Geometry of Moving Images . . . . . 494
*Daniel Fagerström*

Tracking People with a Sparse Network of Bearing Sensors . . . . . 507
*A. Rahimi, B. Dunagan, T. Darrell*

Transformation-Invariant Embedding for Image Analysis . . . . . 519
*Ali Ghodsi, Jiayuan Huang, Dale Schuurmans*

The Least-Squares Error for Structure from Infinitesimal Motion . . . . . 531
*John Oliensis*

Stereo Based 3D Tracking and Scene Learning, Employing Particle Filtering within EM . . . . . 546
*Trausti Kristjansson, Hagai Attias, John Hershey*

## 3D Shape Representation and Reconstruction

The Isophotic Metric and Its Application to Feature Sensitive Morphology on Surfaces . . . . . 560
*Helmut Pottmann, Tibor Steiner, Michael Hofer, Christoph Haider, Allan Hanbury*

A Closed-Form Solution to Non-rigid Shape and Motion Recovery . . . . . 573
*Jing Xiao, Jin-xiang Chai, Takeo Kanade*

Stereo Using Monocular Cues within the Tensor Voting Framework . . . . . 588
*Philippos Mordohai, Gérard Medioni*

Shape and View Independent Reflectance Map from Multiple Views . . . . . 602
*Tianli Yu, Ning Xu, Narendra Ahuja*

**Author Index** . . . . . 617

# A Generic Concept for Camera Calibration

Peter Sturm[1] and Srikumar Ramalingam[2]

[1] INRIA Rhône-Alpes, 38330 Montbonnot, France
Peter.Sturm@inrialpes.fr • http://www.inrialpes.fr/movi/people/Sturm/
[2] Dept. of Computer Science, University of California, Santa Cruz, CA 95064, USA

**Abstract.** We present a theory and algorithms for a generic calibration concept that is based on the following recently introduced general imaging model. An image is considered as a collection of pixels, and each pixel measures the light travelling along a (half-) ray in 3-space associated with that pixel. Calibration is the determination, in some common coordinate system, of the coordinates of all pixels' rays. This model encompasses most projection models used in computer vision or photogrammetry, including perspective and affine models, optical distortion models, stereo systems, or catadioptric systems – central (single viewpoint) as well as non-central ones. We propose a concept for calibrating this general imaging model, based on several views of objects with known structure, but which are acquired from unknown viewpoints. It allows in principle to calibrate cameras of any of the types contained in the general imaging model using one and the same algorithm. We first develop the theory and an algorithm for the most general case: a non-central camera that observes 3D calibration objects. This is then specialized to the case of central cameras and to the use of planar calibration objects. The validity of the concept is shown by experiments with synthetic and real data.

## 1 Introduction

We consider the camera calibration problem, i.e. the estimation of a camera's intrinsic parameters. A camera's intrinsic parameters (plus the associated projection model) give usually exactly the following information: for any point in the image, they allow to compute a ray in 3D along which light travels that falls onto that point (here, we neglect point spread).

Most existing camera models are parametric (i.e. defined by a few intrinsic parameters) and address imaging systems with a single effective viewpoint (all rays pass through one point). In addition, existing calibration procedures are taylor-made for specific camera models.

The aim of this work is to relax these constraints: we want to propose and develop a calibration method that should work for any type of camera model, and especially also for cameras without a single effective viewpoint. To do so, we first renounce on parametric models, and adopt the following very general model: a camera acquires images consisting of pixels; each pixel captures light that travels along a ray in 3D. The camera is fully described by:

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 1–13, 2004.
© Springer-Verlag Berlin Heidelberg 2004

- the coordinates of these rays (given in some local coordinate frame).
- the mapping between rays and pixels; this is basically a simple indexing.

This general imaging model allows to describe virtually any camera that captures light rays travelling along straight lines[1]. Examples (cf. figure 1):

- a camera with any type of optical distortion, such as radial or tangential.
- a camera looking at a reflective surface, e.g. as often used in surveillance, a camera looking at a spherical or otherwise curved mirror [10]. Such systems, as opposed to central catadioptric systems [3] composed of cameras and parabolic mirrors, do not in general have a single effective viewpoint.
- multi-camera stereo systems: put together the pixels of all image planes; they "catch" light rays that definitely do not travel along lines that all pass through a single point. Nevertheless, in the above general camera model, a stereo system (with rigidly linked cameras) is considered as a **single** camera.
- other acquisition systems, see e.g. [4,14,19], insect eyes, etc.

*Relation to previous work.* See [9,17] for reviews and references on existing calibration methods and e.g. [6] for an example related to *central* catadioptric devices. A calibration method for certain types of *non-central* catadioptric cameras (e.g. due to misalignment of mirror), is given in [2].

The above imaging model has already been used, in more or less explicit form, in various works [8,12,13,14,15,16,19,23,24,25], and is best described in [8], were also other issues than sensor geometry, e.g. radiometry, are discussed. There are conceptual links to other works: acquiring an image with a camera of our general model may be seen as sampling the plenoptic function [1], and a light field [11] or lumigraph [7] may be interpreted as a single image, acquired by a camera of an appropriate design.

To our knowledge, the only previously proposed calibration approaches for the general imaging model, are due to Swaminathan, Grossberg and Nayar [8, 22]. The approach in [8] requires the acquisition of two or more images of a calibration object with known structure, and knowledge of the camera or object motion between the acquisitions. In this work, we develop a completely general approach, that requires taking three or more images of calibration objects, from arbitrary and **unknown viewing positions**. The approach in [22] does not require calibration objects, but needs to know the camera motion. Calibration is formulated as a non-linear optimization problem. In this work, "closed-form" solutions are proposed (requiring to solve linear equation systems).

Other related works deal mostly with epipolar geometry estimation and modeling [13,16,24] and motion estimation for already calibrated cameras [12,15].

[1] However, it would not work for example with a camera looking from the air, into water: still, to each pixel is associated a refracted ray in the water. However, when the camera moves, the refraction effect causes the set of rays to move non-rigidly, hence the calibration would be different for each camera position.

*Organization.* In §2, we explain the camera model used and give some notations. For ease of explanation and understanding, the calibration concept is first introduced for 2D cameras, in §3. The general concept for 3D cameras is described in §4 and variants (central vs. non-central camera and planar vs. 3D calibration objects) are developed in §5. Some experimental results are shown in §6, followed by discussions and conclusions.

## 2 Camera Model and Notations

We give the definition of the (purely geometrical) camera model used in this work. It is essentially the same as the model of [8] where in addition other issues such as point spread and radiometry are treated. We assume that a camera delivers images that consist of a set of pixels, where each pixel captures/measures the light travelling along some half-ray. In our calibration method, we do not model half-rays explicitly, but rather use their infinite extensions – **camera rays**. Camera rays corresponding to different pixels need not intersect – in this general case, we speak of **non-central cameras**, whereas if all camera rays intersect in a single point, we have a **central camera** with an **optical center**.

Furthermore, the physical location of the actual photosensitive elements that correspond to pixels, does in general not matter at all. On the one hand, this means that the camera ray corresponding to some pixel, needs not pass through that pixel, cf. figure 1. On the other hand, neighborship relations between pixels are in general not necessary to be taken into account: the set of a camera's photosensitive elements may lie on a single surface patch (image plane), but may also lie on a 3D curve, on several surface patches or even be placed at completely isolated positions. In practice however, we do use some continuity assumption, useful in the stage of 3D-2D matching, as explained in §6: we suppose that pixels are indexed by two integer coordinates like in traditional cameras and that camera rays of pixels with neighboring coordinates, are "close" to one another.

## 3 The Calibration Concept for 2D Cameras

We consider here a camera and scene living in a 2D plane, i.e. camera rays are lines in that plane. Two images are acquired, while the imaged object undergoes some motion. Consider a single pixel and its camera ray, cf. figure 2. Figures 2 (b) and (c) show the two points on the object that are seen by that pixel in the two images. We suppose to be able to determine the coordinates of these two points, in some local coordinate frame attached to the object ("matching").

*The case of known motion.* If the object's motion between image acquisitions is known, then the two object points can be mapped to a single coordinate frame, e.g. the object's coordinate frame at its second position, as shown in figure 2 (d). Computing our pixel's camera ray is then simply done by joining the two points. This summarizes the calibration approach proposed by Grossberg and Nayar [8], applied here for the 2D case. Camera rays are thus initially expressed

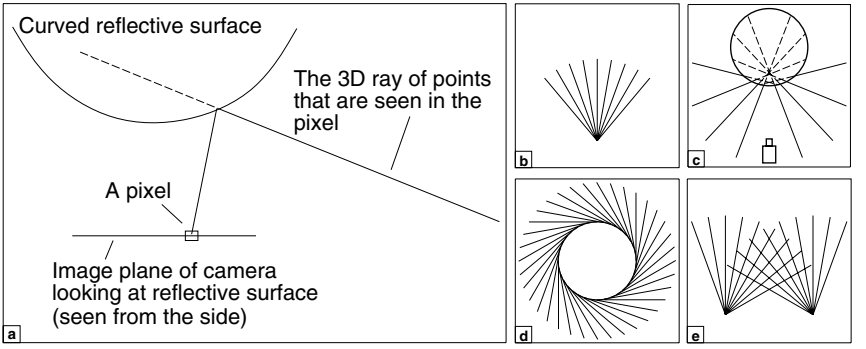



**Fig. 1.** Examples of imaging systems. (a) Catadioptric system. Note that camera rays do not pass through their associated pixels. (b) Central camera (e.g. perspective, with or without radial distortion). (c) Camera looking at reflective sphere. This is a non-central device (camera rays are not intersecting in a single point). (d) Omnivergent imaging system [14,19]. (e) Stereo system (non-central) consisting of two central cameras.

in a coordinate frame attached to the calibration object. This does not matter (all that counts are the relative positions of the rays), but for convenience, one would typically choose a better frame. For a central camera for example, one would choose the optical center as origin or for a non-central camera, the point that minimizes the sum of distances to the set of camera rays (if it exists).

Note that it is not required that the two images be taken of the same object; all that is needed is knowledge of point positions relative to coordinate frames of the objects, and the "motion" between the two coordinate frames.

*The case of unknown motion.* This approach is no longer applicable and we need to estimate, implicitly or explicitly, the unknown motion. We show how to do this, given three images. Let $\mathbf{Q}, \mathbf{Q}'$ and $\mathbf{Q}''$ be the points on the calibration

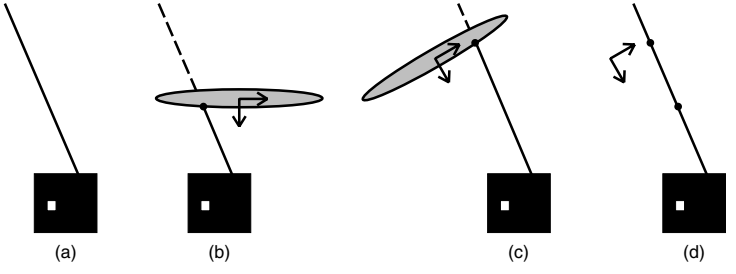



**Fig. 2.** (a) The camera as black box, with one pixel and the associated camera ray. (b) The pixel sees a point on a calibration object, whose coordinates are identified in a frame associated with the object. (c) Same as (b), for another position of the object. (d) Due to known motion, the two points on the calibration object can be placed in the same coordinate frame. The camera ray is then determined by joining them.

objects, that are seen in the same pixel. These are 3-vectors of homogeneous coordinates, expressed in the respective local coordinate frame. Without loss of generality, we choose the coordinate frame associated with the object's first position, as common frame. The unknown relative motions between the second and third frames and the first one, are given by $2 \times 2$ rotation matrices $\mathsf{R}'$ and $\mathsf{R}''$ and translation vectors $\mathbf{t}'$ and $\mathbf{t}''$. Note that $R'_{11} = R'_{22}$ and $R'_{12} = -R'_{21}$ (same for $\mathsf{R}''$). Mapping the calibration points to the common frame gives points

$$\mathbf{Q} \qquad \begin{pmatrix} \mathsf{R}' & \mathbf{t}' \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \mathbf{Q}' \qquad \begin{pmatrix} \mathsf{R}'' & \mathbf{t}'' \\ \mathbf{0}^\mathsf{T} & 1 \end{pmatrix} \mathbf{Q}'' \quad .$$

They must lie on the pixel's camera ray, i.e. must be collinear. Hence, the determinant of the matrix composed of their coordinate vectors, must vanish:

$$\begin{vmatrix} Q_1 & R'_{11}Q'_1 + R'_{12}Q'_2 + t'_1 Q'_3 & R''_{11}Q''_1 + R''_{12}Q''_2 + t''_1 Q''_3 \\ Q_2 & R'_{21}Q'_1 + R'_{22}Q'_2 + t'_2 Q'_3 & R''_{21}Q''_1 + R''_{22}Q''_2 + t''_2 Q''_3 \\ Q_3 & Q'_3 & Q''_3 \end{vmatrix} = 0 \quad . \tag{1}$$

**Table 1.** Non-zero coefficients of the trifocal calibration tensor for a general 2D camera.

| i | $C_i$ | $V_i$ |
|---|---|---|
| 1 | $Q_1 Q'_1 Q''_3 + Q_2 Q'_2 Q''_3$ | $R'_{21}$ |
| 2 | $Q_1 Q'_2 Q''_3 - Q_2 Q'_1 Q''_3$ | $R'_{22}$ |
| 3 | $Q_1 Q'_3 Q''_1 + Q_2 Q'_3 Q''_2$ | $-R''_{21}$ |
| 4 | $Q_1 Q'_3 Q''_2 - Q_2 Q'_3 Q''_1$ | $-R''_{22}$ |
| 5 | $Q_3 Q'_1 Q''_1 + Q_3 Q'_2 Q''_2$ | $R'_{11}R''_{21} - R''_{11}R'_{21}$ |
| 6 | $Q_3 Q'_1 Q''_2 - Q_3 Q'_2 Q''_1$ | $R'_{11}R''_{22} - R''_{12}R'_{21}$ |

| i | $C_i$ | $V_i$ |
|---|---|---|
| 7 | $Q_1 Q'_3 Q''_3$ | $t'_2 - t''_2$ |
| 8 | $Q_2 Q'_3 Q''_3$ | $-t'_1 + t''_1$ |
| 9 | $Q_3 Q'_1 Q''_3$ | $R'_{11}t''_2 - R'_{21}t''_1$ |
| 10 | $Q_3 Q'_2 Q''_3$ | $R'_{12}t''_2 - R'_{22}t''_1$ |
| 11 | $Q_3 Q'_3 Q''_1$ | $R''_{21}t'_1 - R''_{11}t'_2$ |
| 12 | $Q_3 Q'_3 Q''_2$ | $R''_{22}t'_1 - R''_{12}t'_2$ |
| 13 | $Q_3 Q'_3 Q''_3$ | $t'_1 t''_2 - t''_1 t'_2$ |

This equation is trilinear in the calibration point coordinates. The equation's coefficients may be interpreted as coefficients of a trilinear matching tensor; they depend on the unknown motions' coefficients, and are given in table 1. In the following, we sometimes call this the *calibration tensor*. It is somewhat related to the *homography tensor* derived in [18]. Among the $3 \cdot 3 \cdot 3 = 27$ coefficients of the calibration tensor, 8 are always zero and among the remaining 19, there are 6 pairs of identical ones. The columns of table 1 are interpreted as follows: the $C_i$ are trilinear products of point coordinates and the $V_i$ are the associated coefficients of the tensor. The following equation is thus equivalent to (1):

$$\sum_{i=1}^{13} C_i V_i = 0 \quad . \tag{2}$$

Given triplets of points $\mathbf{Q}, \mathbf{Q}'$ and $\mathbf{Q}''$ for at least 12 pixels, we may compute the trilinear tensor up to an unknown scale $\lambda$ by solving a system of linear equations of type (2). Note that we have verified using simulated data, that

we indeed can obtain a unique solution (up to scale) for the tensor. The main problem is then that of extractin the motion parameters from the calibration tensor. In [21] we give a simple algorithm for doing so[2]. Once the motions are determined, the approach described above can be readily applied to compute the camera rays and thus to finalize the calibration.

*The special case of central cameras.* It is worthwhile to specialize the calibration concept to the case of central cameras (but which are otherwise general, i.e. not perspective). A central camera can already be calibrated from two views. Let $\mathbf{Z}$ be the homogeneous coordinates of the optical center (in the frame associated with the object's first position). We have the following collinearity constraint:

$$\begin{vmatrix} Z_1 & Q_1 & R'_{11}Q'_1 + R'_{12}Q'_2 + t'_1Q'_3 \\ Z_2 & Q_2 & R'_{21}Q'_1 + R'_{22}Q'_2 + t'_2Q'_3 \\ Z_3 & Q_3 & Q'_3 \end{vmatrix} = \mathbf{Q}'^{\mathsf{T}} \begin{pmatrix} R'_{21}Z_3 & -R'_{22}Z_3 & R'_{22}Z_2 - R'_{21}Z_1 \\ R'_{22}Z_3 & R'_{21}Z_3 & -R'_{21}Z_2 - R'_{22}Z_1 \\ Z_3t'_2 - Z_2 & Z_1 - Z_3t'_1 & Z_2t'_1 - Z_1t'_2 \end{pmatrix} \mathbf{Q} = 0$$

The bifocal calibration tensor in this equation is a $3 \times 3$ matrix and somewhat similar to a fundamental or essential matrix. It can be estimated linearly from calibration points associated with 8 pixels or more. It is of rank 2 and its right null vector is the optical center $\mathbf{Z}$, which is thus easy to compute. Once this is done, the camera ray for a pixel can be determined e.g. by joining $\mathbf{Z}$ and $\mathbf{Q}$.

*The special case of a linear calibration object.* This is equally worthwhile to investigate. We propose an algorithm in [21], which works but is more complicated than the algorithm for general calibration objects.

## 4 Generic Calibration Concept for 3D Cameras

This and the next section describe our main contributions. We extend the concept described in §3 to the case of cameras living in 3-space. We first deal with the most general case: non-central cameras and 3D calibration objects.

In case of **known motion**, two views are sufficient to calibrate, and the procedure is equivalent to that outlined in §3, cf. [8]. In the following, we consider the practical case of **unknown motion**. Input are now, for each pixel, three 3D points $\mathbf{Q}, \mathbf{Q}'$ and $\mathbf{Q}''$, given by 4-vectors of homogeneous coordinates, relative to the calibration object's local coordinate system. Again, we adopt the coordinate system associated with the first image as global coordinate frame. The object's motion for the other two images is given by $3 \times 3$ rotation matrices $\mathsf{R}'$ and $\mathsf{R}''$ and translation vectors $\mathbf{t}'$ and $\mathbf{t}''$. With the correct motion estimates, the aligned points must be collinear. We stack their coordinates in the following $4 \times 3$ matrix:

$$\begin{pmatrix} Q_1 & R'_{11}Q'_1 + R'_{12}Q'_2 + R'_{13}Q'_3 + t'_1Q'_4 & R''_{11}Q''_1 + R''_{12}Q''_2 + R''_{13}Q''_3 + t''_1Q''_4 \\ Q_2 & R'_{21}Q'_1 + R'_{22}Q'_2 + R'_{23}Q'_3 + t'_2Q'_4 & R''_{21}Q''_1 + R''_{22}Q''_2 + R''_{23}Q''_3 + t''_2Q''_4 \\ Q_3 & R'_{31}Q'_1 + R'_{32}Q'_2 + R'_{33}Q'_3 + t'_3Q'_4 & R''_{31}Q''_1 + R''_{32}Q''_2 + R''_{33}Q''_3 + t''_3Q''_4 \\ Q_4 & Q'_4 & Q''_4 \end{pmatrix} . \quad (3)$$

[2] This is similar, though more complicated than extracting (ego-)motion of perspective cameras from the classical essential matrix [9].

The collinearity constraint means that this matrix must be of rank less than 3, which implies that all sub-determinants of size $3 \times 3$ vanish. There are 4 of them, obtained by leaving out one row at a time. Each of these corresponds to a trilinear equation in point coordinates and thus to a trifocal calibration tensor whose coefficients depend on the motion parameters.

Table 2 gives the coefficients of the first two calibration tensors (all 4 are given in the appendix of [21]). For both, 34 out of 64 coefficients are always zero. One may observe that the two tensors share some coefficients, e.g. $V_8 = W_1 = R'_{31}$.

The tensors can be estimated by solving linear equation system, and we verified using simulated random experiments that in general unique solutions (up to scale) are obtained, if 3D points for sufficiently many pixels (29 at least) are available. In the following, we give an algorithm for computing the motion parameters. Let $V'_i = \lambda V_i$ and $W'_i = \mu W_i, i = 1 \dots 37$ be the estimated tensors (up to scale). The algorithm proceeds as follows.

1. Estimate scale factors: $\lambda = \sqrt{V_8'^2 + V_9'^2 + V_{10}'^2}$ and $\mu = \sqrt{W_1'^2 + W_2'^2 + W_3'^2}$.
2. Compute $V_i = \frac{V'_i}{\lambda}$ and $W_i = \frac{W'_i}{\mu}, i = 1 \dots 37$
3. Compute $\mathsf{R}'$ and $\mathsf{R}''$:

$$\mathsf{R}' = \begin{pmatrix} -W_{15} & -W_{16} & -W_{17} \\ -V_{15} & -V_{16} & -V_{17} \\ V_8 & V_9 & V_{10} \end{pmatrix} \qquad \mathsf{R}'' = \begin{pmatrix} W_{18} & W_{19} & W_{20} \\ V_{18} & V_{19} & V_{20} \\ -V_{11} & -V_{12} & -V_{13} \end{pmatrix} \quad .$$

   They will not be orthonormal in general. We "correct" this as shown in [21].
4. Compute $\mathbf{t}'$ and $\mathbf{t}''$ by solving a straightforward linear least squares problem, which is guaranteed to have a unique solution, see [21] for details.

Using simulations, we verified that the algorithm gives a unique and correct solution in general.

## 5 Variants of the Calibration Concept

Analogously to the case of 2D cameras, cf. §3, we developed important specializations of our calibration concept, for central cameras and planar calibration objects. We describe them very briefly; details are given in [21].

*Central cameras.* In this case, two images are sufficient. Let $\mathbf{Z}$ be the optical center (unknown). By proceeding as in §3, we obtain 4 bifocal calibration tensors of size $4 \times 4$ and rank 2, that are somewhat similar to fundamental matrices. One of them is shown here:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ R'_{31}Z_4 & R'_{32}Z_4 & R'_{33}Z_4 & -Z_3 + Z_4 t'_3 \\ -R'_{21}Z_4 & -R'_{22}Z_4 & -R'_{23}Z_4 & Z_2 - Z_4 t'_2 \\ R'_{21}Z_3 - R'_{31}Z_2 & R'_{22}Z_3 - R'_{32}Z_2 & R'_{23}Z_3 - R'_{33}Z_2 & Z_3 t'_2 - Z_2 t'_3 \end{pmatrix} \quad .$$

It is relatively straightforward to extract the motion parameters and the optical center from these tensors.

**Table 2.** Coefficients of two trifocal calibration tensors for a general 3D camera.

| $i$ | $C_i$ | $V_i$ | $W_i$ |
|---|---|---|---|
| 1 | $Q_1Q'_1Q''_4$ | 0 | $R'_{31}$ |
| 2 | $Q_1Q'_2Q''_4$ | 0 | $R'_{32}$ |
| 3 | $Q_1Q'_3Q''_4$ | 0 | $R'_{33}$ |
| 4 | $Q_1Q'_4Q''_1$ | 0 | $-R''_{31}$ |
| 5 | $Q_1Q'_4Q''_2$ | 0 | $-R''_{32}$ |
| 6 | $Q_1Q'_4Q''_3$ | 0 | $-R''_{33}$ |
| 7 | $Q_1Q'_4Q''_4$ | 0 | $t'_3 - t''_3$ |
| 8 | $Q_2Q'_1Q''_4$ | $R'_{31}$ | 0 |
| 9 | $Q_2Q'_2Q''_4$ | $R'_{32}$ | 0 |
| 10 | $Q_2Q'_3Q''_4$ | $R'_{33}$ | 0 |
| 11 | $Q_2Q'_4Q''_1$ | $-R''_{31}$ | 0 |
| 12 | $Q_2Q'_4Q''_2$ | $-R''_{32}$ | 0 |
| 13 | $Q_2Q'_4Q''_3$ | $-R''_{33}$ | 0 |
| 14 | $Q_2Q'_4Q''_4$ | $t'_3 - t''_3$ | 0 |
| 15 | $Q_3Q'_1Q''_4$ | $-R'_{21}$ | $-R'_{11}$ |
| 16 | $Q_3Q'_2Q''_4$ | $-R'_{22}$ | $-R'_{12}$ |
| 17 | $Q_3Q'_3Q''_4$ | $-R'_{23}$ | $-R'_{13}$ |
| 18 | $Q_3Q'_4Q''_1$ | $R''_{21}$ | $R''_{11}$ |
| 19 | $Q_3Q'_4Q''_2$ | $R''_{22}$ | $R''_{12}$ |
| 20 | $Q_3Q'_4Q''_3$ | $R''_{23}$ | $R''_{13}$ |
| 21 | $Q_3Q'_4Q''_4$ | $t''_2 - t'_2$ | $t''_1 - t'_1$ |
| 22 | $Q_4Q'_1Q''_1$ | $R'_{21}R''_{31} - R''_{21}R'_{31}$ | $R'_{11}R''_{31} - R''_{11}R'_{31}$ |
| 23 | $Q_4Q'_1Q''_2$ | $R'_{21}R''_{32} - R''_{22}R'_{31}$ | $R'_{11}R''_{32} - R''_{12}R'_{31}$ |
| 24 | $Q_4Q'_1Q''_3$ | $R'_{21}R''_{33} - R''_{23}R'_{31}$ | $R'_{11}R''_{33} - R''_{13}R'_{31}$ |
| 25 | $Q_4Q'_1Q''_4$ | $R'_{21}t''_3 - R'_{31}t''_2$ | $R'_{11}t''_3 - R'_{31}t''_1$ |
| 26 | $Q_4Q'_2Q''_1$ | $R'_{22}R''_{31} - R''_{21}R'_{32}$ | $R'_{12}R''_{31} - R''_{11}R'_{32}$ |
| 27 | $Q_4Q'_2Q''_2$ | $R'_{22}R''_{32} - R''_{22}R'_{32}$ | $R'_{12}R''_{32} - R''_{12}R'_{32}$ |
| 28 | $Q_4Q'_2Q''_3$ | $R'_{22}R''_{33} - R''_{23}R'_{32}$ | $R'_{12}R''_{33} - R''_{13}R'_{32}$ |
| 29 | $Q_4Q'_2Q''_4$ | $R'_{22}t''_3 - R'_{32}t''_2$ | $R'_{12}t''_3 - R_{32}t''_1$ |
| 30 | $Q_4Q'_3Q''_1$ | $R'_{23}R''_{31} - R''_{21}R'_{33}$ | $R'_{13}R''_{31} - R''_{11}R'_{33}$ |
| 31 | $Q_4Q'_3Q''_2$ | $R'_{23}R''_{32} - R''_{22}R'_{33}$ | $R'_{13}R''_{32} - R''_{12}R'_{33}$ |
| 32 | $Q_4Q'_3Q''_3$ | $R'_{23}R''_{33} - R''_{23}R'_{33}$ | $R'_{13}R''_{33} - R''_{13}R'_{33}$ |
| 33 | $Q_4Q'_3Q''_4$ | $R'_{23}t''_3 - R'_{33}t''_2$ | $R'_{13}t''_3 - R'_{33}t''_1$ |
| 34 | $Q_4Q'_4Q''_1$ | $R''_{31}t'_2 - R''_{21}t'_3$ | $R''_{31}t'_1 - R''_{11}t'_3$ |
| 35 | $Q_4Q'_4Q''_2$ | $R''_{32}t'_2 - R''_{22}t'_3$ | $R''_{32}t'_1 - R''_{12}t'_3$ |
| 36 | $Q_4Q'_4Q''_3$ | $R''_{33}t'_2 - R''_{23}t'_3$ | $R''_{33}t'_1 - R''_{13}t'_3$ |
| 37 | $Q_4Q'_4Q''_4$ | $t'_2t''_3 - t'_3t''_2$ | $t'_1t''_3 - t''_1t'_3$ |

*Non-central cameras and planar calibration objects.* The algorithm for this case is rather more complicated and not shown here. Using simulations, we proved that we obtain a unique solution in general.

*Central cameras and planar calibration objects.* As with non-central cameras, we already obtain constraints on the motion parameters (and the optical center) from two views of the planar object. In this case however, the associated calibration tensors do not contain sufficient information in order to uniquely estimate the motion and optical center. This is not surprising: even in the very restricted case of **perspective** cameras with 5 intrinsic parameters, two views of a planar calibration object do not suffice for calibration [20,26]. We thus developed an algorithm working with three views [21]. It is rather complicated, but was shown to provide unique solutions in general.

## 6 Experimental Evaluation

As mentioned previously, we verified each algorithm using simulated random experiments. This was first done using noiseless data. We also tested our methods using noisy data and obtained satisfying results. A detailled quantitative analysis remains yet to be carried out.

We did various experiments with real images, using a 3M-Pixel digital camera with moderate optical distortions, a camera with a fish-eye lens and "home-made" catadioptric systems consisting of a digital camera and various curved

off-the-shelf mirrors. We used planar calibration objects consisting of black dots or squares on white paper. Figure 3 shows three views taken by the digital camera.

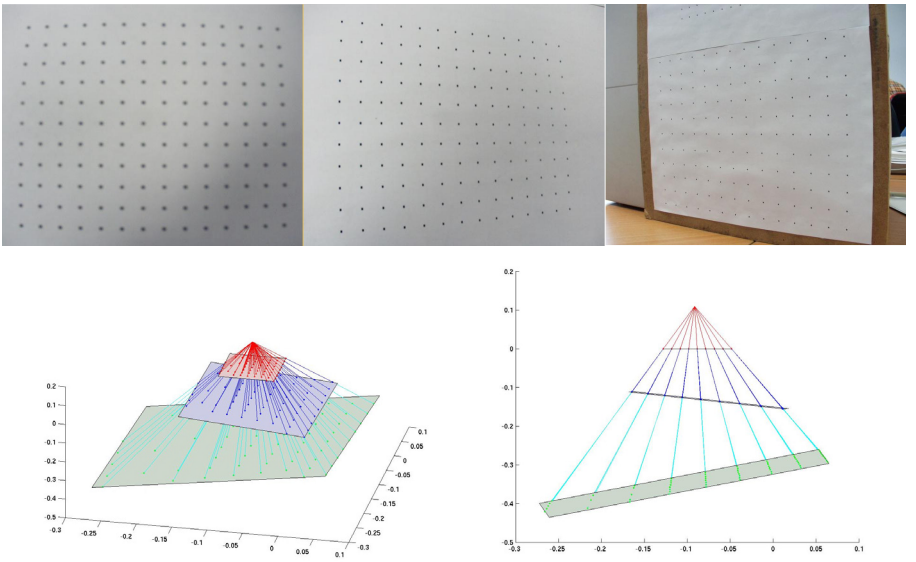


**Fig. 3.** Top: images of 3 boards of different sizes, captured by a digital camera. Bottom: two views of the calibrated camera rays and estimated pose of the calibration boards.

Dots/corners were extracted using the Harris detector. Matching of these image points to points on calibration objects was done semi-automatically. This gives calibration points for a sparse set of pixels per image, and in general there will be few, if any, pixels for which we get a calibration point in every view! We thus take into account the continuity assumption mentioned in §2. For every image, we compute the convex hull of the pixels for which calibration points were extracted. We then compute the intersection of the convex hulls over all three views, and henceforth only consider pixels inside that region. For every such pixel in the first image we estimate the calibration points for the second and third images using the following interpolation scheme: in each of these images, we determine the 4 closest extracted calibration points. We then compute the homography between these pixels and the associated calibration points on the planar object. The calibration point for the pixel of interest is then computed using that homography.

On applying the algorithm for central cameras (cf. §5), we obtained the results shown in figure 3. The bottom row shows the calibrated camera rays and the pose of the calibration objects, given by the estimated motion parameters. It is difficult to evaluate the calibration quantitatively, but we observe that for every

pixel considered, the estimated motion parameters give rise to nearly perfectly collinear calibration points. Note also, cf. the bottom right figure, that radial distortion is correctly modeled: the camera rays are setwise coplanar, although the corresponding sets of pixels in the image are not perfectly collinear.

The same experiment was performed for a fish-eye lens, cf. figure 4. The result is slightly worse – aligned calibration points are not always perfectly collinear. This experiment is preliminary in that only the central image region has been calibrated (cf. figure 4), due to the difficulty of placing planar calibration objects that cover the whole field of view.

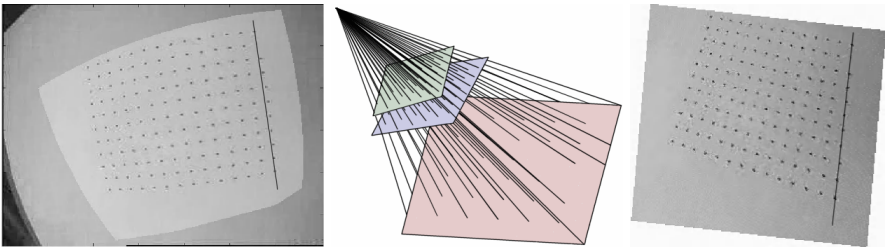


**Fig. 4.** Left: one of 3 images taken by the fish-eye lens (in white the area that was calibrated). Middle: calibrated camera rays and estimated pose of calibration objects. Right: image from the left after distortion correction, see text.

Using the calibration information, we carried out two sample applications, as described in the following. The first one consists in correcting non-perspective distortions: calibration of the central camera model gives us a bunch of rays passing through a single point. We may cut these rays by a plane; at each intersection with a camera ray, we "paint" the plane with the "color" observed by the pixel associated with the ray in some input image. Using the same homography-based interpolation scheme as above, we can thus create a "densely" colored plane, which is nothing else than the image plane of a distortion-corrected perspective image. See figure 4 for an example. This model-free distortion correction scheme is somewhat similar to the method proposed in [5].

Another application concerns (ego-) motion and epipolar geometry estimation. Given calibration information, we can estimate relative camera pose (or motion), and thus epipolar geometry, from two or more views of an unknown object. We developed a motion estimation method similar to [15] and applied it to two views taken by the fish-eye lens. The epipolar geometry of the two views can be computed and visualized as follows: for a pixel in the first view, we consider its camera ray and determine all pixels of the second view whose rays (approximately) intersect the first ray. These pixels form the "epipolar curve" associated with the original pixel. An example is shown in figure 5. The estimated calibration and motion also allow of course to reconstruct objects in 3D (see [21] for examples).

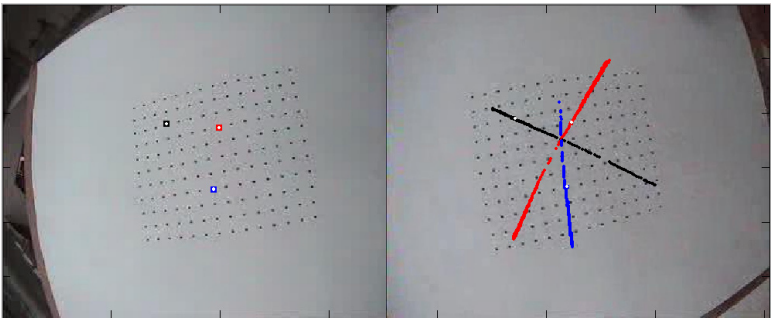


**Fig. 5.** Epipolar curves for three points. These are not straight lines, but intersect in a single point, since we here use the central camera model.

## 7 Discussion

The algorithm for central cameras seems to work fine, even with the minimum input of 3 views and a planar calibration object. Experiments with non-central catadioptric cameras however did so far not give satisfying results. One reason for poor stability of the non-central method is the way we currently obtain our input (homography-based interpolation of calibration points). We also think that the general algorithm, which is essentially based on solving linear equations, can only give stable results with minimum input (3 views) if the considered camera is clearly non-central. By this, we mean that there is not any point that is "close" to all camera rays; the general algorithm does not work for perspective cameras, but for multi-stereo systems consisting of sufficiently many cameras[3].

We propose several ideas for overcoming these problems. Most importantly, we probably need to use several to many images for a stable calibration. We have developed bundle adjustment formulations for our calibration problem, which is not straightforward: the camera model is of discrete nature and does not directly allow to handle sub-pixel image coordinates, which are for example needed in derivatives of a reprojection error based cost function. For initialization of the non-central bundle adjustment, we may use the (stabler) calibration results for the central model. Model selection may be applied to determine if the central or non-central model is more appropriate for a given camera. Another way of stabilizing the calibration might be the possible inclusion of constraints on the set of camera rays, such as rotational or planar symmetry, if appropriate.

Although we have a single algorithm that works for nearly all existing camera types, different cameras will likely require different designs of calibration objects, e.g. panoramic cameras vs. ones with narrow field of view. We stress that a single calibration can use images of different calibration objects; in our experiments, we actually use planar calibration objects of different sizes for the different views, imaged from different distances, cf. figure 3. This way, we can

[3] Refer to the appendix of [21] on the feasibility of the general calibration method for stereo systems consisting of three or more central cameras.

place them such that they do not "intersect" in space, which would give less stable results, especially for camera rays passing close to the intersection region. We also plan to use different calibration objects for initialization and bundle adjustment: initialization, at least for the central model, can be performed using the type of calibration object used in this work. As for bundle adjustment, we might then switch to objects with a much denser "pattern" e.g. with a coating consisting of randomly distributed colored speckles. Another possibility is to use a flat screen to produce a dense set of calibration points [8].

One comment on the difference between calibration and motion estimation: here, with 3 views of a *known* scene, we solve simultaneously for motion and calibration (motion is determined explicitly, calibration implicitly). Whereas once a (general) camera is calibrated, (ego-)motion can already be estimated from 2 views of an *unknown* scene [15]. Hence, although our method estimates motion directly, we consider it a calibration method.

## 8 Conclusions

We have proposed a theory and algorithms for a highly general calibration concept. As for now, we consider this mainly as a conceptual contribution: we have shown how to calibrate nearly any camera, using one and the same algorithm.

We already propose specializations that may be important in practice: an algorithm for central, though otherwise unconstrained cameras, is presented, as well as an algorithm for the use of planar calibration objects. Results of preliminary experiments demonstrate that the approach allows to calibrate central cameras without using any parametric distortion model.

We believe in our concept's potential for calibrating cameras with "exotic" distortions – such as fish-eye lenses with hemispheric field of view or catadioptric cameras, especially non-central ones. We are working towards that goal, by developing bundle adjustment procedures to calibrate from multiple images, and by designing better calibration objects. These issues could bring about the necessary stability to really calibrate cameras without any parametric model in practice. Other ongoing work concerns the extension of classical structure-from-motion tasks such as motion and pose estimation and triangulation, from the perspective to the general imaging model.

## References

1. E.H. Adelson, J.R. Bergen. The Plenoptic Function and the Elements of Early Vision. *Computational Models of Visual Processing*, MIT Press, 1991.
2. D.G. Aliaga. Accurate Catadioptric Calibration for Real-time Pose Estimation in Room-size Environments. *ICCV*, 127-134, 2001.
3. S. Baker, S. Nayar. A Theory of Catadioptric Image Formation. *ICCV*, 1998.
4. H. Bakstein, T. Pajdla. An overview of non-central cameras. *Proceedings of Computer Vision Winter Workshop*, Ljubljana, Slovenia, 2001.
5. P. Brand. Reconstruction tridimensionnelle d'une scène à partir d'une caméra en mouvement.PhD Thesis, Université Claude Bernard, Lyon, October 1995.

6. C. Geyer, K. Daniilidis. Paracatadioptric Camera Calibration. *PAMI*, 2002.
7. S.J. Gortler et al.The Lumigraph. *SIGGRAPH*, 1996.
8. M.D. Grossberg, S.K. Nayar. A general imaging model and a method for finding its parameters. *ICCV*, 2001.
9. R.I. Hartley, A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
10. R.A. Hicks, R. Bajcsy. Catadioptric Sensors that Approximate Wide-angle Perspective Projections. *CVPR*, pp. 545-551, 2000.
11. M. Levoy, P. Hanrahan. Light field rendering. *SIGGRAPH*, 1996.
12. J. Neumann, C. Fermüller, Y. Aloimonos. Polydioptric Camera Design and 3D Motion Estimation. *CVPR*, 2003.
13. T. Pajdla. Stereo with oblique cameras. *IJCV*, 47(1), 2002.
14. S. Peleg, M. Ben-Ezra, Y. Pritch. OmniStereo: Panoramic Stereo Imaging. *PAMI*, pp. 279-290, March 2001.
15. R. Pless. Using Many Cameras as One. *CVPR*, 2003.
16. S. Seitz. The space of all stereo images. *ICCV*, 2001.
17. C.C. Slama (editor). *Manual of Photogrammetry*. Fourth Edition, ASPRS, 1980.
18. A. Shashua, L. Wolf. Homography Tensors: On Algebraic Entities That Represent Three Views of Static or Moving Planar Points. *ECCV*, 2000.
19. H.-Y. Shum, A. Kalai, S.M. Seitz. Omnivergent Stereo. *ICCV*, 1999.
20. P. Sturm, S. Maybank. On Plane-Based Camera Calibration. *CVPR*, 1999.
21. P. Sturm, S. Ramalingam. A Generic Calibration Concept: Theory and Algorithms. Research Report 5058, INRIA, France, 2003.
22. R. Swaminathan, M.D. Grossberg, and S.K. Nayar. Caustics of Catadioptric Cameras. *ICCV*, 2001.
23. R. Swaminathan, M.D. Grossberg, S.K. Nayar. A perspective on distortions. *CVPR*, 2003.
24. Y. Wexler, A.W. Fitzgibbon, A. Zisserman. Learning epipolar geometry from image sequences. *CVPR*, 2003.
25. D. Wood et al.Multiperspective panoramas for cell animation. *SIGGRAPH*, 1997.
26. Z. Zhang. A flexible new technique for camera calibration. *PAMI*, 22(11), 2000.

# General Linear Cameras

Jingyi Yu[1,2] and Leonard McMillan[2]

[1] Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA,
jingyi@graphics.lcs.mit.edu
[2] Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27911, USA
mcmillan@cs.unc.edu

**Abstract.** We present a General Linear Camera (GLC) model that unifies many previous camera models into a single representation. The GLC model is capable of describing all perspective (pinhole), orthographic, and many multiperspective (including pushbroom and two-slit) cameras, as well as epipolar plane images. It also includes three new and previously unexplored multiperspective linear cameras. Our GLC model is both general and linear in the sense that, given any vector space where rays are represented as points, it describes all 2D affine subspaces (planes) that can be formed by affine combinations of 3 rays. The incident radiance seen along the rays found on subregions of these 2D affine subspaces are a precise definition of a projected image of a 3D scene. The GLC model also provides an intuitive physical interpretation, which can be used to characterize real imaging systems. Finally, since the GLC model provides a complete description of all 2D affine subspaces, it can be used as a tool for first-order differential analysis of arbitrary (higher-order) multiperspective imaging systems.

## 1 Introduction

Camera models are fundamental to the fields of computer vision and photogrammetry. The classic pinhole and orthographic camera models have long served as the workhorse of 3D imaging applications. However, recent developments have suggested alternative multiperspective camera models [4,20] that provide alternate and potentially advantageous imaging systems for understanding the structure of observed scenes. Researchers have also recently shown that these multiperspective cameras are amenable to stereo analysis and interpretation [13, 11,20].

In contrast to pinhole and orthographic cameras, which can be completely characterized using a simple linear model (the classic 3 by 4 matrix [5]), multiperspective cameras models are defined less precisely. In practice, multiperspective cameras models are described by constructions. By this we mean that a system or process is described for generating each specific class. While such physical models are useful for both acquisition and imparting intuition, they are not particularly amenable to analysis.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 14–27, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this paper we present a unified General Linear Camera (GLC) model that is able to describe nearly all useful imaging systems. In fact, under an appropriate interpretation, it describes all possible linear images. In doing so it provides a single model that unifies existing perspective and multiperspecive cameras.

## 2 Previous Work

The most common linear camera model is the classic 3 x 4 pinhole camera matrix [5], which combines six extrinsic and five intrinsic camera parameters into single operator that maps homogenous 3D points to a 2D image plane. These mappings are unique down to a scale factor, and the same infrastructure can also be used to describe orthographic cameras. Recently, several researchers have proposed alternative camera representations known as multiperspective cameras which capture rays from different points in space. These multiperspective cameras include pushbroom cameras [4], which collect rays along parallel planes from points swept along a linear trajectory, and two-slit cameras [10], which collect all rays passing through two lines. Zomet et al [20] did an extensive analysis and modelling of two slit(XSlit) multiperspective cameras. However, they discuss the relationship of these cameras to pinhole cameras only for the purpose of image construction, whereas we provide a unifying model.

Multiperspective imaging has also been explored in the field of computer graphics. Examples include multiple-center-of-projection images [12], manifold mosaics [11], and multiperspective panoramas [18]. Most multiperspective images are generated by stitching together parts of pinhole images [18,12], or slicing through image sequences [11,20].

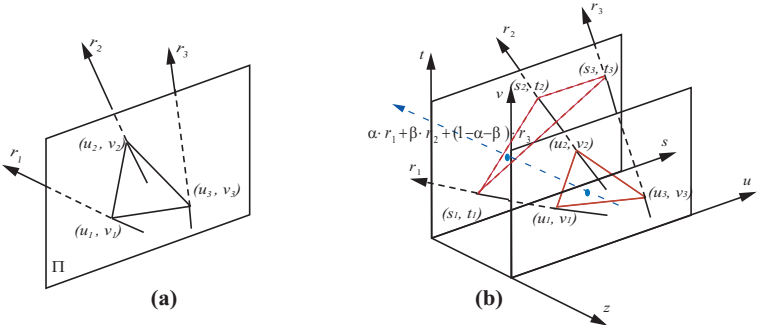



**Fig. 1.** General Linear Camera Model. a)A GLC is characterized by three rays originated from the image plane. b)It collects all possible affine combination of three rays.

Seitz [13] has analyzed the space of multiperspective cameras to determine those with a consistent epipolar geometry. Their work suggests that some multiperspective images can be used to analyze three-dimensional structure, just as

pinhole cameras are commonly used. We focus our attention on a specific class of linear multiperspective cameras, most of which satisfy Seitz's criterion.

Our analysis is closely related to the work of Gu et al [3], which explored the linear structures of 3D rays under a particular 4D mapping known as a two-plane parametrization. This model is commonly used for light field rendering. Their primary focus was on the duality of points and planes under this mapping. They deduced that XSlits are another planar structure within this space, but they do not characterize all of the possible planar structures, nor discuss their analogous camera models.

Our new camera model only describes the set of rays seen by a particular camera, not their distribution on the image plane. Under this definition pinhole cameras are defined by only 3 parameters (the position of the pinhole in 3D). Homographies and other non-linear mappings of pinhole images (i.e., radial distortion) only change the distribution of rays in the image plane, but do not change the set of rays seen. Therefore, all such mappings are equivalent under our model.

## 3 General Linear Camera Model

The General Linear Camera (GLC) is defined by three rays that originate from three points $p_1(u_1, v_1)$, $p_2(u_2, v_2)$ and $p_3(u_3, v_3)$ on an image plane $\Pi_{image}$, as is shown in Figure 1. A GLC collects radiance measurements along all possible "affine combinations" of these three rays. In order to define this affine combination of rays, we assume a specific ray parametrization.

W.o.l.g, we define $\Pi_{image}$ to lie on $z = 0$ plane and its origin to coincide with the origin of the coordinate system. From now on, we call $\Pi_{image}$ as $\Pi_{uv}$. In order to parameterize rays, we place a second plane $\Pi_{st}$ at $z = 1$. All rays not parallel to $\Pi_{st}, \Pi_{uv}$ will intersect the two planes at $(s, t, 1)$ and $(u, v, 0)$ respectively. That gives a 4D parametrization of each ray in form $(s, t, u, v)$. This parametrization for rays, called the two-plane parametrization (2PP), is widely used by the computer graphics community for representing light fields and lumigraphs [7,2]. Under this parametrization, an affine combination of three rays $r_i(s_i, t_i, u_i, v_i)$, $i = 1, 2, 3$, is defined as:

$$r = \alpha \cdot (s_1, t_1, u_1, v_1) + \beta \cdot (s_2, t_2, u_2, v_2) + (1 - \alpha - \beta) \cdot (s_3, t_3, u_3, v_3) \qquad (1)$$

The choice of $\Pi_{st}$ at $z = 1$, is, of course, arbitrary. One can choose any plane parallel to $\Pi_{uv}$ to derive an equivalent parametrization. Moreover, these alternate parameterizations will preserve affine combinations of three rays.

**Lemma 1.** *The affine combinations of any three rays under two different 2PP parameterizations that differ by choice of $\Pi_{st}$ (i.e., $(s, t, u, v)$ and $(s', t', u, v)$ ) are the same.*

*Proof.* Suppose $\Pi_{s't'}$ is at some arbitrary depth $z_0$, $z_0 \neq 0$. Consider the transformation of a ray between the default parametrization ($z_0 = 1$) and this new one.

If $r(s,t,u,v)$ and $r(s',t',u,v)$ represent the same ray $r$ in 3D, then $r(s,t,u,v)$ must pass through $(s',t',z_0)$, and there must exist some $\lambda$ such that

$$\lambda \cdot (s,t,1) + (1-\lambda) \cdot (u,v,0) = (s',t',z_0) \tag{2}$$

Solving for $\lambda$, we have

$$s' = s \cdot z_0 + u \cdot (1-z_0), \quad t' = t \cdot z_0 + v \cdot (1-z_0) \tag{3}$$

Since this transformation is linear, and affine combinations are preserved under linear transformation, the affine combinations of rays under our default two-plane parametrization ($z_0 = 1$) will be consistent for parameterizations over alternative parallel planes. Moreover, the affine weights for a particular choice of parallel $\Pi_{st}$ are general. □

We call the GLC model "linear" because it defines all 2-dimensional affine subspaces in the 4-dimensional "ray space" imposed by a two-plane parametrization. Moreover, these 2D affine subspaces of rays can be considered as images. We refer to the three rays used in a particular GLC as the GLC's *generator rays*. Equivalently, a GLC can be described by the coordinates of two triangles with corresponding vertices, one located on $\Pi_{st}$, and the second on $\Pi_{uv}$. Unless otherwise specified, we will assume the three generator rays (in their 4D parametrization) are linearly independent. This affine combination of generator rays also preserves linearity, while other parameterizations, such as the 6D *Plücker* coordinates [16], do not [3].

**Lemma 2.** *If three rays are parallel to a plane $\Pi$ in 3D, then all affine combinations of them are parallel to $\Pi$ as well.*

**Lemma 3.** *If three rays intersect a line $l$ parallel to the image plane, all affine combinations of them will intersect $l$ as well.*

*Proof.* By lemma 1, we can reparametrize three rays by placing $\Pi_{st}$ so that it contains $l$ resulting in the same set of affine combinations of the three rays. Because the $st$ plane intersections of the three rays will lie on $l$, all affine combinations of three rays will have their $st$ coordinates on $l$, i.e., they will all pass through $l$. The same argument can be applied to all rays which pass through a given point. □

## 4 Equivalence of Classic Camera Models

Traditional camera models have equivalent GLC representations.

**Pinhole camera:** By definition, all rays of a pinhole camera pass through a single point, C in 3D space (the center of projection). Any three linearly independent rays from C will the intersect $\Pi_{uv}$ and $\Pi_{st}$ planes to form two triangles. These triangles will be similar and have parallel corresponding edges,

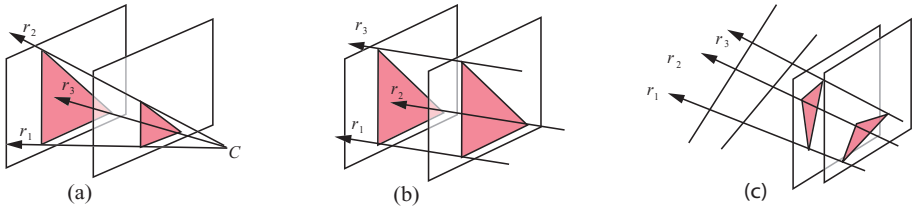



**Fig. 2.** Classic camera models represented as GLC. (a)Two similar triangles on two planes define a pinhole camera; (b)Two parallel congruent triangles define an orthographic camera; (c)Three rays from an XSlit camera.

as shown in Figure 2(a). Furthermore, any other ray, $r$, through C will intersect $\Pi_{uv}$ and $\Pi_{st}$ planes at points $\dot{p}_{uv}$, and $\dot{q}_{st}$. These points will have the same affine coordinates relative to the triangle vertices on their corresponding planes, and $r$ has the same affine coordinates as these two points.

**Orthographic camera:** By definition, all rays on an orthographic camera have the same direction. Any three linearly independent rays from an orthographic camera intersect parallel planes at the vertices of congruent triangles with parallel corresponding edges, as shown in Figure 2(b). Rays connecting the same affine combination of these triangle vertices, have the same direction as the 3 generator rays, and will, therefore, originate from the same orthographic camera.

**Pushbroom camera:** A pushbroom camera sweeps parallel planes along a line $l$ collecting those rays that pass through $l$. We refer to this family of parallel planes as $\Pi^*$. We choose $\Pi_{uv}$ parallel to $l$ but not containing $l$, and select a non-degenerate set of generator rays (they intersect $\Pi_{uv}$ in a triangle). By Lemma 2 and 3, all affine combinations of the three rays must all lie on $\Pi^*$ parallel planes and must also pass through $l$ and, hence, must belong to the pushbroom camera. In the other direction, for any point $\dot{p}$ on $\Pi_{uv}$, there exist one ray that passes through $\dot{p}$, intersects $l$ and is parallel to $\Pi^*$. Since $\dot{p}$ must be some affine combination of the three vertexes of the $uv$ triangle, $r$ must lie on the corresponding GLC. Furthermore, because all rays of the pushbroom camera will intersect $\Pi_{uv}$, the GLC must generate equivalent rays.

**XSlit camera:** By definition, an XSlit camera collects all rays that pass through two non-coplanar lines. We choose $\Pi_{uv}$ to be parallel to both lines but to not contain either of them. One can then pick a non-degenerate set of generator rays and find their corresponding triangles on $\Pi_{st}$ and $\Pi_{uv}$. By Lemma 3, all affine combinations of these three rays must pass through both lines and hence must belong to the XSlit camera. In the other direction, authors of XSlit [10,20] have shown that each point $\dot{p}$ on the image plane $\Pi_{uv}$, maps to a unique ray $r$ in an XSlit camera. Since $\dot{p}$ must be some affine combination of the three vertexes of the $uv$ triangle, $r$ must belong to the GLC. The GLC hence must generate equivalent rays as the XSlit camera.

**Epipolar Plane Image:** EPI [1] cameras collect all rays that lie on a plane in 3D space. We therefore can pick any three linearly independent rays on the

plane as generator rays. Affine combinations of these rays generate all possible rays on the plane,so long as they are linearly independent. Therefore a GLC can also represent Epipolar Plane Images.

## 5 Characteristic Equation of GLC

Although we have shown that a GLC can represent most commonly used camera models, the representation is not unique (i.e., three different generator rays can define the same camera). In this section we develop a criterion to classify general linear cameras. One discriminating characteristic of affine ray combinations is whether or not all rays pass through a line in 3D space. This characteristic is fundamental to the definition of many multi-perspective cameras. We will use this criteria to define the characteristic equation of general linear cameras.

Recall that any 2D affine subspace in 4D can be defined as affine combinations of three points. Thus, GLC models can be associated with all possible planes in the 4D since GLCs are specified as affine combinations of three rays, whose duals in 4D are the three points.

**Lemma 4.** *Given a non-EPI, non-pinhole GLC, if all camera rays pass through some line $l$, not at infinity, in 3D space, then $l$ must be parallel to $\Pi_{uv}$.*

*Proof.* We demonstrate the contrapositive. If $l$ is not parallel to $\Pi_{uv}$, and all rays on a GLC pass through $l$, then we show the GLC must be either an EPI or a pinhole camera.

Assume the three rays pass through at least two distinct points on $l$, otherwise, they will be on a pinhole camera, by Lemma 3. If $l$ is not parallel, then it must intersect $\Pi_{st}$, $\Pi_{uv}$ at some point $(s_0, t_0, 1)$ and $(u_0, v_0, 0)$. Gu et al [3] has shown all rays passing through $l$ must satisfy the following bilinear constraints

$$(u - u_0)(t - t_0) - (v - v_0)(s - s_0) = 0 \tag{4}$$

We show that the only GLCs that satisfy this constraint are EPIs or pinholes.

All 2D affine subspaces in $(s, t, u, v)$ can be written as the intersection of two linear constraints $A_i \cdot s + B_i \cdot t + C_i \cdot u + D_i \cdot v + E_i = 0$, $i = 1, 2$. In general we can solve these two equations for two variables, for instance, we can solve for $u$-$v$ as

$$u = A'_1 \cdot s + B'_1 \cdot t + E'_1 \ , \ v = A'_2 \cdot s + B'_2 \cdot t + E'_2 \tag{5}$$

Substituting $u$ and $v$ into the bilinear constraint (4), we have

$$(A'_1 \cdot s + B'_1 \cdot t + E'_1 - u_0)(t - t_0) = (A'_2 \cdot s + B'_2 \cdot t + E'_2 - v_0)(s - s_0) \tag{6}$$

This equation can only be satisfied for all $s$ and $t$ if $A'_1 = B'_2$ and $B'_1 = A'_2 = 0$, therefore, equation (5) can be rewritten as $u = A' \cdot s + E'_1$ and $v = A' \cdot t + E'_2$. Gu et al [3] have shown all rays in this form must pass through a 3D point $P$ ($P$ cannot be at infinity, otherwise all rays have uniform directions and cannot all pass through any line $l$, not at infinity). Therefore all rays must lie on a 3D

plane that passes through $l$ and finite $P$. The only GLC camera in which all rays lie on a 3D plane is an EPI. If the two linear constraints are singular in $u$ and $v$, we can solve for $s$-$t$, and similar results hold.

If the two linear constraints cannot be solved for $u$-$v$ or $s$-$t$ but can be solved for $u$-$s$ or $v$-$t$, then a similar analysis results in equations of two parallel lines, one on $\Pi_{st}$, the other on $\Pi_{uv}$. The set of rays through two parallel lines must lie on an EPI. □

Lemma 3 and 4 imply that given a GLC, we need only consider if the three generator rays pass through some line parallel to $\Pi_{st}$. We use this relationship to define the characteristic equation of a GLC.

The three generator rays in a GLC correspond to the following 3D lines:

$$r_i = \lambda_i \cdot (s_i, t_i, 1) + (1 - \lambda_i) \cdot (u_i, v_i, 0) \quad i = 1, 2, 3$$

The three rays intersect some plane $\Pi_{z=\lambda}$ parallel to $\Pi_{uv}$ when $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$. By Lemma 3, all rays on the GLC pass through some line $l$ on $\Pi_{z=\lambda}$ if the three generator rays intersect $l$. Therefore, we only need to test if there exist any $\lambda$ so that the three intersection points of the generator rays with $\Pi_{z=\lambda}$ lie on a line. A necessary and sufficient condition for 3 points on a constant $z$-plane to be co-linear is that they have zero area on that plane. This area is computed as follows (Note: the value of $z$ is unnecessary):

$$\begin{vmatrix} (\lambda \cdot s_1 + (1-\lambda) \cdot u_1) & (\lambda \cdot t_1 + (1-\lambda) \cdot v_1) & 1 \\ (\lambda \cdot s_2 + (1-\lambda) \cdot u_2) & (\lambda \cdot t_2 + (1-\lambda) \cdot v_2) & 1 \\ (\lambda \cdot s_3 + (1-\lambda) \cdot u_3) & (\lambda \cdot t_3 + (1-\lambda) \cdot v_3) & 1 \end{vmatrix} = 0 \tag{7}$$

Notice equation (7) is a quadratic equation in $\lambda$ of the form

$$A \cdot \lambda^2 + B \cdot \lambda + C = 0 \tag{8}$$

where

$$A = \begin{vmatrix} s_1 - u_1 & t_1 - v_1 & 1 \\ s_2 - u_2 & t_2 - v_2 & 1 \\ s_3 - u_3 & t_3 - v_3 & 1 \end{vmatrix}, \; B = \begin{vmatrix} s_1 & v_1 & 1 \\ s_2 & v_2 & 1 \\ s_3 & v_3 & 1 \end{vmatrix} - \begin{vmatrix} t_1 & u_1 & 1 \\ t_2 & u_2 & 1 \\ t_3 & u_3 & 1 \end{vmatrix} - 2 \cdot \begin{vmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ u_3 & v_3 & 1 \end{vmatrix}, \; C = \begin{vmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ u_3 & v_3 & 1 \end{vmatrix}$$

We call equation (8) the characteristic equation of a GLC. Since the characteristic equation can be calculated from any three rays, one can also evaluate the characteristic equation for EPI and pinhole cameras. The number of solutions of the characteristic equation implies the number of lines that all rays on a GLC pass through. It may have 0, 1, 2 or infinite solutions. The number of solutions depends on the denominator $A$ and the quadratic discriminant $\Delta = B^2 - 4AC$.

We note that the characteristic equation is invariant to translations in 4D. Equivalently, translations of the two triangles formed by generator rays $(s_i', t_i') = (s_i + T_s, t_i + T_t)$, $(u_i', v_i') = (u_i + T_u, v_i + T_v)$, $i = 1, 2, 3$, do not change the coefficients $A$, $B$ and $C$ of equation (8).

## 6 Characterizing Classic Camera Models

In this section, we show how to identify standard camera models using the characteristic equation of 3 given generator rays.

**Lemma 5.** *Given a GLC, three generator rays, and its characteristic equation* $A \cdot \lambda^2 + B \cdot \lambda + C = 0$*, then all rays are parallel to some plane if and only if* $A = 0$*.*

*Proof.* Notice in the matrix used to calculate $A$, row $i$ is the direction $\boldsymbol{d}_i$ of ray $r_i$. Therefore $A$ can be rewritten as $A = (\boldsymbol{d}_1 \times \boldsymbol{d}_2) \cdot \boldsymbol{d}_3$. Hence $A = 0$ if and only if $\boldsymbol{d}_1$, $\boldsymbol{d}_2$ and $\boldsymbol{d}_3$ are parallel to some 3D plane. And by Lemma 2, all affine combinations of these rays must also be parallel to that plane if $A = 0$. □

### 6.1 $A = 0$ Case

When $A = 0$, the characteristic equation degenerates to a linear equation, which can have 1, 0, or an infinite number of solutions. By Lemma 5, all rays are parallel to some plane. Only three standard camera models satisfy this condition: pushbroom, orthographic, and EPI.

All rays of a pushbroom lie on parallel planes and pass through one line, as is shown in Figure 4(a). A GLC is a pushbroom camera if and only if $A = 0$ and the characteristic equation has 1 solution.

All rays of an orthographic camera have the same direction and do not all simultaneously pass through any line $l$. Hence its characteristic equation has no solution. The zero solution criteria alone, however, is insufficient to determine if a GLC is orthographic. We show in the following section that one can *twist* an orthographic camera into bilinear sheets by rotating rays on parallel planes, as is shown in Figure 4(b), and still maintain that all rays do not pass through a common line. In Section 3, we showed that corresponding edges of the two congruent triangles of an orthographic GLC must be parallel. This parallelism is captured by the following expression:

$$\frac{(s_i - s_j)}{(t_i - t_j)} = \frac{(u_i - u_j)}{(v_i - v_j)} \quad i, j = 1, 2, 3 \;\; \textit{and} \;\; i \neq j \tag{9}$$

We call this condition the *edge-parallel* condition. It is easy to verify that a GLC is orthographic if and only if $A = 0$, its characteristic equation has no solution, and it satisfies the *edge-parallel* condition.

Rays of an EPI camera all lie on a plane and pass through an infinite number of lines on the plane. In order for a characteristic equation to have infinite number of solutions when $A = 0$, we must also have $B = 0$ and $C = 0$. This is not surprising, because the intersection of the epipolar plane with $\Pi_{st}$ and $\Pi_{uv}$ must be two parallel lines and it is easy to verify $A = 0$, $B = 0$ and $C = 0$ if and only if the corresponding GLC is an EPI.

### 6.2 $A \neq 0$ Case

When $A \neq 0$, the characteristic equation becomes quadratic and can have 0, 1, or 2 solutions, which depends on the characteristic equation's discriminant $\Delta$. We show how to identify the remaining two classical cameras, pinhole and XSlit cameras in term of $A$ and $\Delta$.

All rays in a pinhole camera pass through the center of projection (COP). Therefore, any three rays from a pinhole camera, if linearly independent, cannot all be parallel to any plane, and by Lemma 4, $A \neq 0$. Notice that the roots of the characteristic equation correspond to *the depth* of the line that all rays pass through, hence the characteristic equation of a pinhole camera can only have one solution that corresponds to the depth of the COP, even though there exists an infinite number of lines passing through the COP. Therefore, the characteristic equation of a pinhole camera must satisfy $A \neq 0$ and $\Delta = 0$. However, this condition alone is insufficient to determine if a GLC is pinhole. In the following section, we show that there exists a camera where all rays lie on pencil of planes sharing a line, as shown in Figure 4(c), which also satisfies these conditions. One can, however, reuse the *edge-parallel* condition to verify if a GLC is pinhole. Thus a GLC is pinhole, if and only if $A \neq 0$, has one solution, and it satisfies *edge-parallel* condition.

Rays of an XSlit camera pass through two slits and, therefore, the characteristic equation of a GLC must have at least two distinct solutions. Furthermore, Pajdla [10] has shown all rays of an XSlit camera cannot pass through lines other than its two slits, therefore, the characteristic equation of an XSlit camera has exactly two distinct solutions. Thus, a GLC is an XSlit if and only if $A \neq 0$ and $\Delta > 0$.

## 7 New Multiperspective Camera Models

The characteristic equation also suggests three new multiperspective camera types that have not been previously discussed. They include 1)twisted orthographic: $A = 0$, the equation has no solution, and all rays do not have uniform direction; 2)pencil camera: $A \neq 0$ and the equation has one root, but all rays do not pass through a 3D point; 3)bilinear camera: $A \neq 0$ and the characteristic equation has no solution. In this section, we give a geometric interpretation of these three new camera models.

Before describing these camera models, however, we will first discuss a helpful interpretation of the spatial relationships between the three generator rays. An affine combination of two 4D points defines a 1-dimensional affine subspace. Under 2PP, a 1-D affine subspaces corresponds to a bilinear surface $S$ in 3D that contains the two rays associated with each 4D point. If these two rays intersect or have the same direction in 3D space, $S$ degenerates to a plane. Next, we consider the relationship between ray $r_3$ and $S$. We define $r_3$ to be parallel to $S$ if and only if $r_3$ has the same direction as some ray $r \in S$. This definition of parallelism is quite different from conventional definitions. In particular, if $r_3$

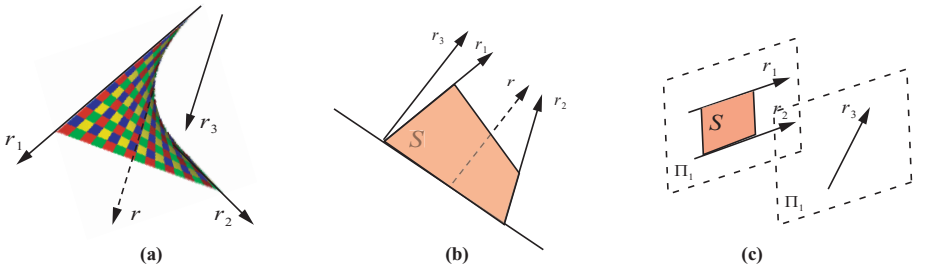



**Fig. 3.** Bilinear Surfaces. (a) $r_3$ is parallel to $S$; (b) $r_3$ is parallel to $S$, but still intersects $S$; (c) $r_3$ is not parallel to $S$, and does not intersect $S$ either.

is parallel to $S$, $r_3$ can still intersect $S$. And if $r_3$ is not parallel to $S$, $r_3$ still might not intersect $S$, Figure 3(b) and (c) show examples of each case.

This definition of parallelism, however, is closely related to $A$ in the characteristic equation. If $r_3$ is parallel to S, by definition, the direction of $r_3$ must be some linear combination of the directions of $r_1$ and $r_2$, and, therefore, $A = 0$ by Lemma 5. $A = 0$, however, is not sufficient to guarantee $r_3$ is parallel to $S$. For instance, one can pick two rays with uniform directions so that $A = 0$, yet still have the freedom to pick a third so that it is not parallel to the plane, as is shown in Figure 3(c).

The number of solutions to the characteristic equation is also closely related to the number of intersections of $r_3$ with $S$. If $r_3$ intersects the bilinear surface $S(r_1, r_2)$ at $P$, then there exists a line $l$, where $P \in l$, that all rays pass through. This is because one can place a constant-z plane that passes through P and intersects $r_1$ and $r_2$ at $Q$ and $R$. It is easy to verify that $P$, $Q$ and $R$ lie on a line and, therefore, all rays must pass through line $PQR$. Hence $r_3$ intersecting $S(r_1, r_2)$ is a sufficient condition to ensure that all rays pass through some line. It further implies if the characteristic equation of a GLC has no solution, no two rays in the camera intersect. GLCs whose characteristic equation has no solution are examples of the oblique camera from [9].

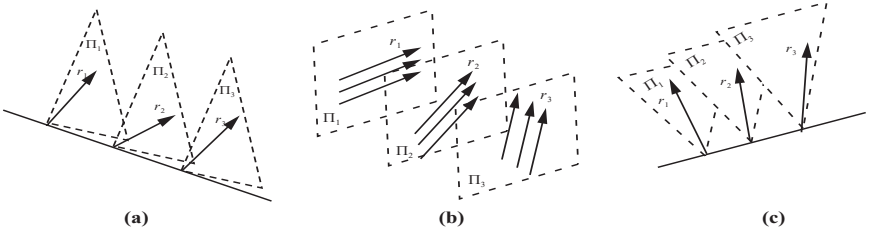



**Fig. 4.** Pushbroom, Twisted Orthographic, and Pencil Cameras. (a) A pushbroom camera collects rays on a set of parallel planes passing through a line; (b) A twisted orthographic camera collects rays with uniform directions on a set of parallel planes; (c) A pencil camera collects rays on a set of non-parallel planes that share a line.

### 7.1 New Camera Models

Our GLC model and its characteristic equation suggests 3 new camera types that have not been previously described.

**Twisted Orthographic Camera:** The characteristic equation of the twisted orthographic camera satisfies $A = 0$, has no solution, and its generators do not satisfy the *edge-parallel* condition. If $r_1$, $r_2$ and $r_3$ are linearly independent, no solution implies $r_3$ will not intersect the bilinear surface $S$. In fact, no two rays intersect in 3D space. In addition, $A = 0$ also implies that all rays are parallel to some plane $\Pi$ in 3D space, therefore the rays on each of these parallel planes must have uniform directions as is shown in Figure 4(b). Therefore, twisted orthographic camera can be viewed as twisting parallel planes of rays in an orthographic camera along common bilinear sheets.

**Pencil Camera:** The characteristic equation of a pencil camera satisfies $A \neq 0$, has one solution and the generators do not satisfy the *edge-parallel* condition. In Figure 4(c), we illustrate a sample pencil camera: rays lie on a pencil of planes that share line $l$. In a pushbroom camera, all rays also pass through a single line. However, pushbroom cameras collect rays along planes transverse to $l$ whereas the planes of a pencil camera contains $l$ (i.e., lie in the pencil of planes through $l$), as is shown in Figure 4(a) and 4(c).

**Bilinear Camera:** By definition, the characteristic equation of a bilinear camera satisfies $A \neq 0$ and the equation has no solution ($\Delta < 0$). Therefore, similar to twisted orthographic cameras, no two rays intersect in 3D in a bilinear camera. In addition, since $A \neq 0$, no two rays are parallel either. Therefore, any two rays in a bilinear camera form a non-degenerate bilinear surface, as is shown in Figure 3(a). The complete classification of cameras is listed in Table 1.

**Table 1.** Characterize General Linear Cameras by Characteristic Equation

| Characteristic Equation | 2 Solution | 1 Solution | 0 Solution | Inf. Solution |
|---|---|---|---|---|
| $A \neq 0$ | XSlit | Pencil/Pinhole† | Bilinear | Ø |
| $A = 0$ | Ø | Pushbroom | Twisted/Ortho.† | EPI |

†: A GLC satisfying *edge-parallel* condition is pinhole($A \neq 0$) or orthographic ($A = 0$).

### 7.2 All General Linear Cameras

Recall that the characteristic equation of a GLC is invariant to translation, therefore we can translate $(s_1, t_1)$ to $(0, 0)$ to simplify computation. Furthermore, we assume the $uv$ triangle has canonical coordinates $(0, 0)$, $(1, 0)$ and $(0, 1)$. This gives:

$$A = s_2 t_3 - s_3 t_2 - s_2 - t_3 + 1, \quad \Delta = (s_2 - t_3)^2 + 4 s_3 t_2 \tag{10}$$

The probability that $A = 0$ is very small, therefore, pushbroom, orthographic and twisted orthographic cameras are a small subspace of GLCs. Furthermore since

$s_2$, $t_2$, $s_3$ and $t_3$ are independent variables, we can, by integration, determine that approximately two thirds of all possible GLCs are XSlit, one third are bilinear cameras, and remainders are other types.

## 8 Example GLC Images

In Figure 5, we compare GLC images of a synthetic scene. The distortions of the curved isolines on the objects illustrate various multi-perspective effects of GLC cameras. In Figure 6, we illustrate GLC images from a 4D light field. Each GLC is specified by three generator rays shown in red. By appropriately transforming the rays on the image plane via a 2D homography, most GLCs generate easily interpretable images. In Figure 7, we choose three desired rays from different pinhole images and fuse them into a multiperspective bilinear GLC image.

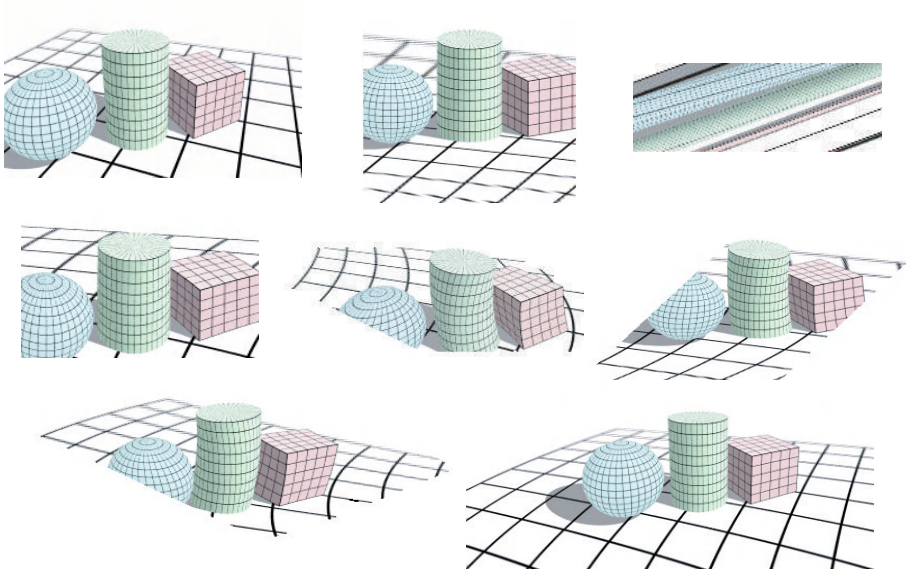


**Fig. 5.** Comparison between synthetic GLC images. From left to right, top row: a pinhole, an orthographic and an EPI; middle row: a pushbroom, a pencil and an twisted orthographic; bottom row: a bilinear and an XSlit.

## 9 Conclusions

We have presented a General Linear Camera (GLC) model that unifies perspective (pinhole), orthographic and many multiperspective (including pushbroom and two-slit) cameras, as well as Epipolar Plane Images (EPI). We have also

**Fig. 6.** GLC images created from a light field. Top row: a pencil, bilinear, and push-broom image. Bottom row: an XSlit, twisted orthographic, and orthographic image.

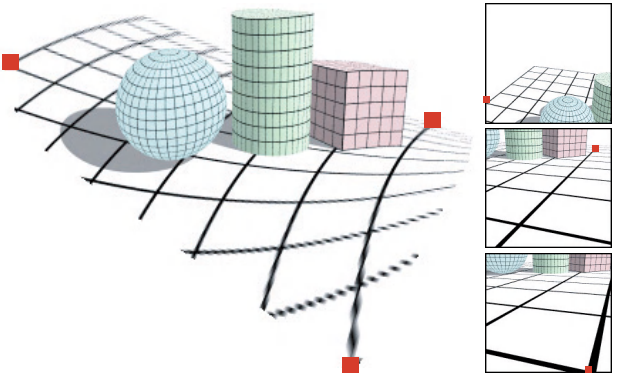


**Fig. 7.** A multiperspective bilinear GLC image synthesized from three pinhole cameras shown on the right. The generator rays are highlighted in red.

introduced three new linear multiperspective cameras that have not been previously explored: they are twisted orthographic, pencil and bilinear cameras. We have further deduced the characteristic equation for every GLC from its three generator rays and have shown how to use it to classify GLCs into eight canonical camera models.

The GLC model also provides an intuitive physical interpretation between lines, planar surfaces and bilinear surfaces in 3D space, and can be used to characterize real imaging systems like mirror reflections on curved surface. Since GLCs describes all possible 2D affine subspaces in 4D ray space, they can used be as a tool for first-order differential analysis of these high-order multiperspective imaging systems. GLC images can be rendered directly by ray tracing a synthetic scene, or by cutting through pre-captured light fields. By appropriately organizing rays, all eight canonical GLCs generate interpretable images similar

to pinhole and orthographic cameras. Furthermore, we have shown one can fuse desirable features from different perspectives to form any desired multiperspective image.

## References


1. Bolles, R. C., H. H. Baker, and D. H. Marimont: Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion. International Journal of Computer Vision, Vol. 1 (1987).
2. S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen: The Lumigraph. Proc. ACM SIGGRAPH '96 (1996) 43–54.
3. Xianfeng Gu, Steven J. Gortler, and Michael F. Cohen. Polyhedral geometry and the two-plane parameterization. Eurographics Rendering Workshop 1997 (1997) pages 1–12.
4. R. Gupta and R.I. Hartley: Linear Pushbroom Cameras. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 9 (1997) 963–975.
5. R.I. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge Univ. Press, 2000.
6. R. Kingslake, Optics in Photography. SPIE Optical Eng., Press, 1992.
7. M. Levoy and P. Hanrahan: Light Field Rendering. Proc. ACM SIGGRAPH '96 (1996) 31–42.
8. B. Newhall, The History of Photography, from 1839 to the Present Day. The Museum of Modern Art (1964) 162.
9. T. Pajdla: Stereo with Oblique Cameras. Int'l J. Computer Vision, vol. 47, nos. 1/2/3 (2002) 161–170.
10. T. Pajdla: Geometry of Two-Slit Camera. Research Report CTU–CMP–2002–02, March 2002.
11. S. Peleg, M. Ben-Ezra, and Y. Pritch: Omnistereo: Panoramic Stereo Imaging. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 23, no. 3 (2001) 279–290.
12. P.Rademacher and G.Bishop: Multiple-center-of-Projection Images. Proc. ACM SIGGRAPH '98 (1998)199–206.
13. S.M. Seitz: The Space of All Stereo Images. Proc. Int'l Conf. Computer Vision '01, vol. I (2001) 26–33.
14. J. Semple and G. Kneebone: Algebraic Projective Geometry. Oxford: Clarendon Press, 1998.
15. H.-Y. Shum, A. Kalai, and S. M. Seitz: Omnivergent stereo. In Proc. 7th Int. Conf. on Computer Vision (1999) 22–29.
16. D. Sommerville, Analytical Geometry of Three Dimensions. Cambridge University Press, 1959.
17. T. Takahashi, H. Kawasaki, K. Ikeuchi, and M. Sakauchi: Arbitrary View Position and Direction Rendering for Large-Scale Scenes. Proc. IEEE Conf. Computer Vision and Pattern Recognition (2000) 296–303.
18. D. Wood, A. Finkelstein, J. Hughes, C. Thayer, and D. Salesin: Multiperspective Panoramas for Cel Animation. Proc. ACM SIGGRAPH '97 (1997) 243-250.
19. J.Y. Zheng and S. Tsuji: Panoramic Representation for Route Recognition by a Mobile Robot. Int'l J. Computer Vision, vol. 9, no. 1 (1992) 55–76.
20. A. Zomet, D. Feldman, S. Peleg, and D. Weinshall: Mosaicing New Views: The Crossed-Slits Projection. IEEE Trans. on PAMI (2003) 741–754.

# A Framework for Pencil-of-Points Structure-from-Motion

Adrien Bartoli[1,2], Mathieu Coquerelle[2], and Peter Sturm[2]

[1] Department of Engineering Science, University of Oxford, UK
[2] équipe MOVI, INRIA Rhône-Alpes, France
Bartoli@robots.ox.ac.uk, Coquerelle@inria.fr, Sturm@inria.fr

**Abstract.** Our goal is to match contour lines between images and to recover structure and motion from those. The main difficulty is that pairs of lines from two images do not induce direct geometric constraint on camera motion. Previous work uses geometric attributes — orientation, length, etc. — for single or groups of lines. Our approach is based on using Pencil-of-Points (points on line) or POPs for short. There are many advantages to using POPs for structure-from-motion. The most important one is that, contrarily to pairs of lines, pairs of POPs may constrain camera motion. We give a complete theoretical and practical framework for automatic structure-from-motion using POPs — detection, matching, robust motion estimation, triangulation and bundle adjustment. For wide baseline matching, it has been shown that cross-correlation scores computed on neighbouring patches to the lines gives reliable results, given 2D homographic transformations to compensate for the pose of the patches. When cameras are known, this transformation has a 1-dimensional ambiguity. We show that when cameras are unknown, using POPs lead to a 3-dimensional ambiguity, from which it is still possible to reliably compute cross-correlation. We propose linear and non-linear algorithms for estimating the fundamental matrix and for the multiple-view triangulation of POPs. Experimental results are provided for simulated and real data.

## 1 Introduction

Recovering structure and motion from images is one of the key goals in computer vision. A common approach is to detect and match image features while recovering camera motion. The goal of this paper is the automatic matching of lines and recovery of structure and motion. This problem is difficult for the reason that a pair of corresponding lines does not give direct geometric constraint on the camera motion. Hence, one has to work on a three-view basis or assume that camera motion is known a priori, e.g. [10].

In this paper, we attack directly the two view case by introducing a type of image primitive that we call *Pencil-of-Points* or POP for short. A POP is made of a *supporting line* and a set of *supporting points* lying on the supporting line. Physically, a POP corresponds to a set of interest points on a contour line. POPs can be built on the top of most contour lines. Contrarily to pairs of corresponding

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 28–40, 2004.
© Springer-Verlag Berlin Heidelberg 2004

lines, pairs of corresponding POPs may give geometric constraints on camera motion, provided that what we call the *local geometry*, relating corresponding points along the supporting lines, has been computed. We exploit these geometric constraints for matching POPs and recovering structure and motion. Once camera motion has been recovered using POPs, it can be employed for a reliable guided-matching and reconstruction of other types of features.

The closest work to ours is [10]. The main difference is that the authors consider that the cameras are known and propose a wide-baseline guided-matching algorithm for lines. They show that reliable results are obtained based on cross-correlation scores, computed by warping the neighbouring textures of the lines using the 2D homography $\mathsf{H}(\mu) \sim [\mathbf{l}']_\times \mathsf{F} + \mu \mathbf{e}' \mathbf{l}^\mathsf{T}$, where $\mathbf{l} \leftrightarrow \mathbf{l}'$ are corresponding lines, $\mathsf{F}$ is the fundamental matrix and $\mathbf{e}'$ the second epipole. The projective parameter $\mu$ is computed by minimizing the cross-correlation score.

Before going into further details about our approach, we underline some of the advantages of using POPs for automatic structure and motion recovery. First, a POP has fewer degrees of freedom than the supporting line and the individual supporting points which implies that (*i*) its localization is often more accurate that those of the individual features, (*ii*) finding POPs in a set of interest points and contour lines increase their individual repeatability rate and (*iii*) structure and motion parameters estimated from POPs are more accurate than that recovered from points and/or lines. Second, matching or tracking POPs through images is more reliable than individual contour lines or interest points, since a pair of corresponding POPs defines a local geometry, used to score matching hypotheses based on geometric or photometric criteria. Third, the robust estimation of camera motion based on random sampling from putative correspondences, i.e. in a RANSAC-like manner [3], is more efficient using POPs than other standard features, since only three pairs of POPs define a fundamental matrix, versus seven pairs of points.

*Contributions and paper organization.* Using POPs for structure-from-motion is a new concept. We propose a comprehensive framework for multiple-view matching and recovery of structure and motion. Our framework is based on the following traditional steps, which also give the organization of this paper.

First, §2, we investigate the detection of POPs in images and their matching. We define and study the *local geometry* of a pair of POPs. We propose methods for its estimation, which allow to obtain putative POP correspondences, from which the epipolar geometry can be robustly estimated.

Second, §3, we propose techniques for estimating the epipolar geometry from POP correspondences. Minimal and redundent cases are studied.

Third, §4, we tackle the problem of triangulating POPs from multiple images. We derive and approximate the optimal (in the Maximum Likelihood sens) solution by an algorithm based on the triangulation of the supporting line, then the supporting points.

Finally, bundle adjustment is described in §5. We provide experimental results on simulated data and give our conclusions and further work in §§6 and 7 respectively. Experimental results on real data are provided throughout the

paper. The following two paragraphs give our notation, some preliminaries and definitions.

*Notation and preliminaries.* We make no formal distinction between coordinate vectors and physical entities. Equality up to a non-null scale factor is denoted by $\sim$. Vectors are typeset using bold font ($\mathbf{q}$, $\mathbf{Q}$), matrices using sans-serif fonts ($\mathsf{F}$, $\mathsf{H}$) and scalars in italic ($\alpha$). Transposition and transposed inverse are denoted by $^{\mathsf{T}}$ and $^{-\mathsf{T}}$. The $(3 \times 3)$ skew-symmetric cross-product matrix is written as in $[\mathbf{q}]_{\times}\mathbf{x} = \mathbf{q} \times \mathbf{x}$. Indices are used to indicate the size of a matrix or vector ($\mathsf{F}_{(3\times3)}$, $\mathbf{q}_{(3\times1)}$), to index a set of entities ($\mathbf{q}_i$) or to select coefficients of matrices or vectors ($q_1$, $q_{i,1}$). Index $i$ is used for the $n$ images, $j$ for the $m$ features and $k$ for the $p$ supporting points of a POP[1]. The supporting lines are written $\mathbf{l}_{ij}$ (the supporting line of the $j$-th POP in image $i$) and supporting points as $\mathbf{q}_{ijk}$ (the $k$-th supporting point of the $j$-th POP in image $i$). Indices are sometimes dropped for clarity. The identity matrix is written I and the null-vector as $\mathbf{0}$. We use the Euclidean distance between points, denoted $d_e$ and an algebraic distance defined by:

$$d_a^2(\mathbf{q}, \mathbf{u}) = \|\mathsf{S}[\mathbf{q}]_{\times}\mathbf{u}\|^2 \quad \text{with} \quad \mathsf{S} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{1}$$

*Definitions.* A pencil of points is a set of $p$ supporting points lying on a supporting line. If $p \geq 3$, the POP is said to be *complete*, otherwise, it is said to be *incomplete*. A *complete correspondence* is a correspondence of complete POPs. As shown in the next section, only complete correspondences may define a local geometry.

We distinguish two kinds of correspondences of POPs: *line-level* and *point-level* correspondences. A line-level correspondence means that only the supporting lines are known to match. A point-level correspondence is stronger and means that a point-to-point mapping along the supporting lines has been established.

## 2 Detecting and Matching Pencil-of-Points

### 2.1 Detecting

Detecting POPs in images is the first step of the structure-from-motion process. One of the most important properties of a detector is its ability to achieve repeatability rates[2] as high as possible, which reflects the fact that it can detect the same features in different images. In order to ensure high repeatability rates, we formulate our POP detector based on interest points and contour lines, for which there exist detectors achieving high repeatability rates, see [9] for interest points and [2] for contour lines.

In order to detect salient POPs, we merge nearby contour lines. Algorithms based on the Hough transform or RANSAC [3] can be used to detect POPs within

[1] To simplify the notation, we assume without loss of generality that all POPs have the same number of supporting points.

[2] The repeatability rate between two images is the number of corresponding features over the mean number of detected points [9].

a set of points and/or lines. We propose the following simple solution. First, an empty POP is instanciated for each line (which gives the supporting line). Second, each point is attached to the POPs whose supporting line is at a distance lower than a threshold, that we typically choose as a few pixels. Finally, incomplete POPs, i.e. those for which the number of supporting points is less than three, are eliminated. Note that we use a loose threshold for interest point and contour line detection, to get as many as possible POPs. The less significant interest points and contour lines are generally pruned as they are respectively not attached to any POP or form incomplete POPs. An example of POP detection is shown on figures 1 (a) & (b). It is observed that the repeatability rate of POPs is higher than each of the repeatability rates of points and lines.

**Fig. 1.** (a) & (b) Show the detected POPs. The repeatability rate is 51% while for points and lines it is lower, respectively 41% and 37%. (c) & (d) show the 9 putative matches obtained with our algorithm. On this example, all of them are correct, which shows the robustness of our local geometry based cross-correlation measure.

### 2.2 Matching

Traditional structure-from-motion algorithms using interest points usually rely on an initial matching, followed by the robust estimation of camera geometry and a guided-matching step, see e.g. [6]. The initial matching step is often based on similarity measures between points such as correlation or grey-value invariants. Guided-matching uses the estimated camera geometry to constrain the search-area. In the case of POPs, the initial matching step is based on the local geometry defined by a pair of POPs. This step is described below followed by the robust estimation of the epipolar geometry.

**Matching Based on Local Geometry.** As mentioned above, the idea is to use the local geometry defined by a pair of POPs. We show that this local geometry is modeled by a 1D homography and allows to establish dense correspondences

between the two supporting lines. Given a hypothesized line-level POP correspondence, we upgrade it to point-level by computing its local geometry. Given a point-level correspondence, a similarity score can be computed using cross-correlation, in a manner similar to [10]. For each POP in one image, the score is computed for all POPs in the other image and a 'winner takes all' scheme is employed to extract a set of putative POP matches. Putative matches obtained by our algorithm are shown on figures 1 (c) & (d).

*Defining and computing the local geometry.* We study the local geometry induced by a point-level correspondence, and propose an estimation method.

**Proposition 1.** *Corresponding supporting points are linked by a 1D homography, related to the* epipolar transformation*, relating corresponding epipolar lines.*

*Proof:* Corresponding supporting points lie on corresponding epipolar lines: there is a trivial one-to-one correspondence between supporting points and epipolar lines (provided the supporting lines do not contain the epipoles). The proof follows from the fact that the epipolar pencils are related by a 1D homography [12]. ■

First, we shall define a local $\mathbb{P}^1$ parameterization of the supporting points, using two Euclidean transformation matrices $\mathsf{A}$ and $\mathsf{A}'$ acting such that the supporting lines are rotated to be vertical and aligned with the $y$-axes of the images. The transformed supporting points are $\mathbf{x}_k \sim \mathsf{A}\mathbf{q}_k \sim (0 \;\; y_k \;\; 1)^\mathsf{T}$ and $\mathbf{x}'_k \sim \mathsf{A}'\mathbf{q}'_k \sim (0 \;\; y'_k \;\; 1)^\mathsf{T}$. Second, we introduce a 1D homography $\mathsf{g}$ as:

$$\begin{pmatrix} y'_k \\ 1 \end{pmatrix} \sim \mathsf{g} \begin{pmatrix} y_k \\ 1 \end{pmatrix} \quad \text{with} \quad \mathsf{g} \sim \begin{pmatrix} g_1 & g_2 \\ g_3 & 1 \end{pmatrix}, \tag{2}$$

which is equivalent to $\mathbf{x}' \sim \mathsf{G}(\boldsymbol{\mu})\mathbf{x}$ with $\mathsf{G}(\boldsymbol{\mu}) \sim \begin{pmatrix} \mu_1 & 0 & 0 \\ \mu_2 & g_1 & g_2 \\ \mu_3 & g_3 & 1 \end{pmatrix}$, where the 3-vector $\boldsymbol{\mu}^\mathsf{T} \sim (\mu_1 \;\; \mu_2 \;\; \mu_3)$ represents projective parameters which are significant only when $\mathsf{G}(\boldsymbol{\mu})$ is applied to points off the supporting line. The 2D homography mapping corresponding points along the supporting lines is $\mathsf{H}(\boldsymbol{\mu}) \sim \mathsf{A}'^{-1}\mathsf{G}(\boldsymbol{\mu})\mathsf{A}$.

The 1D homography $\mathsf{g}$ can be estimated from $p \geq 3$ pairs of supporting points using equation (2). This is the reason why complete POPs are defined as those which have at least 3 supporting points. Given $\mathsf{g}$, $\mathsf{H}(\boldsymbol{\mu})$ can be formed.

*Computing* $\mathsf{H}(\boldsymbol{\mu})$. The above-described algorithm can not be applied directly since at this stage, we only have line-level POP correspondence hypotheses. We have to upgrade them to point-level to estimate $\mathsf{H}(\boldsymbol{\mu})$ with the previously-given algorithm and score them by computing cross-correlation. We propose the following algorithm:

- for all valid pairs of triplets of supporting points[3]:
  - compute the local geometry represented by $\mathsf{H}(\boldsymbol{\mu})$.
  - compute the cross-correlation score based on $\mathsf{H}(\boldsymbol{\mu})$, see below.
- return the $\mathsf{H}(\boldsymbol{\mu})$ corresponding to the highest cross-correlation score.

[3] Valid triplets satisfy an ordering constraint, namely middle points have to match.

*Computing cross-correlation.* For a pair of POPs, the matching score is obtained by evaluating the cross-correlation using $\mathsf{H}(\boldsymbol{\mu})$ to associate corresponding points. The cross-correlation is evaluated within rectangular strips centered onto the supporting lines. The length of the strips are given by the overlap of the supporting lines in each image. The width of the strips must be sufficiently large for cross-correlation to be discriminative. During our experiments, we found that a width of 3 to 7 pixels was appropriate. For pixels off the supporting lines, the $\boldsymbol{\mu}$ parameters are significant. The following solutions are possible: compute these parameters by minimizing the cross-correlation score, as in [10], or use the median luminance and chrominance of the regions adjacent to the supporting lines [1]. The first solution is computationally too expensive to be used in our inner loop, since 3 parameters have to be estimated, while the second solution is not discriminative enough. We propose to map pixels along lines perpendicular to the supporting lines. Hence, the method uses neighbouring texture while being independent of $\boldsymbol{\mu}$. In order to take into account a possible non-planarity surrounding the supporting lines, we weight the contribution of each pixel to cross-correlation proportionally to the inverse of its distance to the supporting line.

**Robustly Computing the Epipolar Geometry.** At this stage, we are given a set of putative POP correspondences. We employ a robust estimator, allowing to estimate the epipolar geometry and to discriminate between inliers and outliers. We use a scheme based on RANSAC [3], which maximizes the number of inliers. In order to use RANSAC, one must provide a *minimal estimator*, i.e. an estimator which computes the epipolar geometry from the minimum number of correspondences, and a function to discriminate between inliers and outliers, given an hypothesized epipolar geometry. The number of trials required to ensure a good probability of success, say 0.99, depends on the minimal number of correspondences needed to compute the epipolar geometry. Our minimal estimator described in §3 needs 3 pairs of POPs. Applying a RANSAC procedure is therefore much more efficient with POPs than with points: with 50% of outliers, 35 trials are sufficient with POPs, while 588 trials are required for points (values taken from [6]).

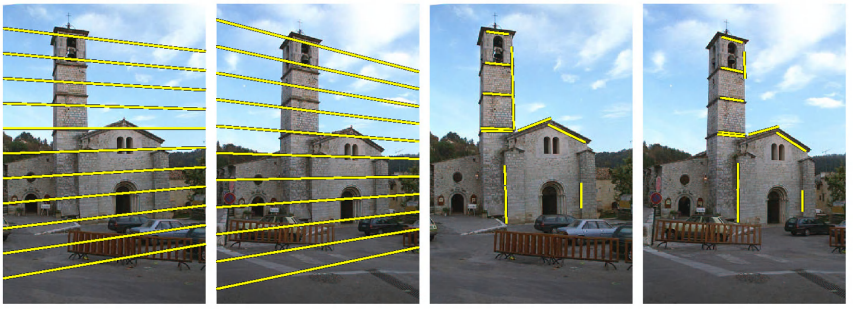
Our inlier/outlier discriminating function is based on computing the cross-correlation score using [10]. Inliers are selected by thresholding this score. We use a threshold of few percents (2% — 5%) of the maximal grey value. Figures 2 (a-d) show an example of epipolar geometry computation, and the set of corresponding POPs obtained after guided-matching based on the method of [10].

## 3 Computing the Epipolar Geometry

**Proposition 2.** *The minimal number of pairs of* POP*s in general position*[4] *needed to define a unique fundamental matrix is 3.*

*Proof:* Due to lack of space, this proof is left for an extended version of the paper.

[4] General position means that the supporting lines are not coplanar and do not lie on an epipolar plane, i.e. the image lines do not contain the epipoles.



(a) (b) (c) (d)

**Fig. 2.** (a) & (b) Show a representative set of corresponding epipolar lines while (c) & (d) show the 11 matched lines obtained after guided-matching using the algorithm of [10].

### 3.1 The 'Eight Corrected Point' Algorithm

This linear estimator is based on the constraints induced by the supporting points. Pairs of supporting points $\mathbf{q}_{jk} \leftrightarrow \mathbf{q}'_{jk}$ are obtained based on the previously estimated local geometries $\mathsf{H}(\boldsymbol{\mu})$. The first idea that comes to mind is to use the supporting points as input to the eight point algorithm [7]. This algorithm minimizes an algebraic distance between predicted epipolar lines and observed points. The eight corrected point algorithm consists in correcting the position of the supporting points, i.e. to make them colinear, prior to applying the eight point algorithm. Using this procedure reduces the noise on the points positions, as we shall verify experimentally.

### 3.2 The 'Three POP' Algorithm

This linear algorithm compares observed points and predicted points. This algorithm is more statistically meaningful than the eight point algorithm, in the case of POPs, in that observed and predicted features are directly compared.

We wish to predict the supporting point positions. We intersect the predicted epipolar lines, i.e. $\mathsf{F}\mathbf{q}_{jk}$ in the second image, with the supporting lines $\mathbf{l}'_j$: the predicted point is given by $[\mathbf{l}'_j]_\times \mathsf{F}\mathbf{q}_{jk}$. Our cost function is given by summing the squared algebraic distances between observed and predicted points: $\sum_j d_a^2(\mathbf{q}'_{jk}, [\mathbf{l}'_j]_\times \mathsf{F}\mathbf{q}_{jk})$. In order to obtain a symmetric criterion, we consider predicted and observed points in the first image also, which yields:

$$\mathcal{C}_a = \sum_j \sum_k \left( d_a^2(\mathbf{q}_{jk}, [\mathbf{l}_j]_\times \mathsf{F}^\mathsf{T} \mathbf{q}'_{jk}) + d_a^2(\mathbf{q}'_{jk}, [\mathbf{l}'_j]_\times \mathsf{F}\mathbf{q}_{jk}) \right). \tag{3}$$

After introducing explicitly $d_a$ from equation (1) and minor algebraic manipulations, we obtain the matrix form $\mathcal{C}_a = \sum_j \sum_k (\|\mathsf{B}_{jk}\mathbf{f}\|^2 + \|\mathsf{B}'_{jk}\mathbf{f}\|^2)$ where $\mathbf{f} = \text{vect}(\mathsf{F})$ is the row-wise vectorization of $\mathsf{F}$ and:

$$\mathsf{B}_{jk} = \mathsf{S}[\mathbf{q}_{jk}]_\times [\mathbf{l}_j]_\times \left(q'_{jk,1}\mathrm{I}\ \ q'_{jk,2}\mathrm{I}\ \ q'_{jk,3}\mathrm{I}\right), \mathsf{B}'_{jk} = \mathsf{S}[\mathbf{q}'_{jk}]_\times [\mathbf{l}'_j]_\times \mathrm{diag}(\mathbf{q}_{jk}^\mathsf{T}\ \ \mathbf{q}_{jk}^\mathsf{T}\ \ \mathbf{q}_{jk}^\mathsf{T}).$$

The cost function becomes $\mathcal{C}_a = \|\mathsf{B}\mathbf{f}\|^2$ with $\mathsf{B}^\mathsf{T} \sim \left(\mathsf{B}_{11}^\mathsf{T}\ \ \mathsf{B}'_{11}{}^\mathsf{T}\ \ \ldots\ \ \mathsf{B}_{mp}^\mathsf{T}\ \ \mathsf{B}'_{mp}{}^\mathsf{T}\right)$. The singular vector associated to the smallest singular value of $\mathsf{B}$ gives the $\mathbf{f}$ that minimizes $\mathcal{C}_a$. Similarly to the eight point algorithm, the obtained fundamental matrix does not satisfy the rank-deficiency constraint in general, and has to be corrected by nullifying its smallest singular value, see e.g. [6].

### 3.3 Non-linear 'Reduced' Estimation

The previously-described three POP estimator is statistically sound in the sense that observed and predicted points are compared in the linear cost function (3). However, the comparison is done using the algebraic distance $d_a$. This is the price to pay to get a linear estimator. In this section, we consider a cost function with a similar form, but using the Euclidean distance $d_e$ to compare observed and predicted points:

$$\mathcal{C}_e = \sum_j \sum_k \left(d_e^2(\mathbf{q}_{jk}, [\mathbf{l}_j]_\times \mathsf{F}^\mathsf{T}\mathbf{q}'_{jk}) + d_e^2(\mathbf{q}'_{jk}, [\mathbf{l}'_j]_\times \mathsf{F}\mathbf{q}_{jk})\right). \tag{4}$$

We use the Levenberg-Marquart algorithm, see e.g. [6], with a suitable parameterization of the fundamental matrix [12] to minimize this cost function, based on the initial solution provided by the three POP algorithm.

## 4 Multiple-View Triangulation

We deal with the triangulation of POP seen in multiple views. Note that since the triangulation of a line is independent from the others, we drop the index $j$ in this section.

### 4.1 Optimal Triangulation

The optimal 3D POP is the one which better explains the data, i.e. which minimizes the sum of squared Euclidean distances between predicted and observed supporting points. Assuming that 3D POPs are represented by two points $\mathbf{M}$ and $\mathbf{N}$ for the supporting line and $p$ scalars $\alpha_k$ for the supporting points $\mathbf{Q}_k \sim \alpha_k\mathbf{M} + (1-\alpha_k)\mathbf{N}$, the following non-linear problem is obtained:

$$\min_{\mathbf{M},\mathbf{N},\ldots,\alpha_k,\ldots} \mathcal{C}_{pop} \text{ with } \mathcal{C}_{pop} = \sum_{i=1}^{n}\sum_{k=1}^{p} d_e^2(\mathsf{P}_i(\alpha_k\mathbf{M} + (1-\alpha_k)\mathbf{N}), \mathbf{q}_{ik}). \tag{5}$$

We use the Levenberg-Marquart algorithm, e.g. [6]. We examine the difficult problem of finding a reliable initial solution in the next section.

### 4.2 Initialization

Finding an initial solution which is close to the optimal one is of primary importance. The initialization method must minimize a cost function as close as possible to (5). We propose a two-step initialization algorithm consisting in triangulating the supporting line, then each supporting point. Our motivations for these steps are explained while reviewing line triangulation below.

**Line Triangulation.** Line triangulation from multiple views is a standard structure-from-motion problem and has been widely studied, see e.g. [5]. The optimal line $< \mathbf{M}, \mathbf{N} >$ is given by minimizing the sum of squared Euclidean distances between the predicted lines $(\mathsf{P}_i\mathbf{M}) \times (\mathsf{P}_i\mathbf{N})$ and the observed points $\mathbf{q}_{ik}$ as $\min_{\mathbf{M},\mathbf{N}} \sum_{i=1}^{n} \sum_{k=1}^{p} d_e^2((\mathsf{P}_i\mathbf{M}) \times (\mathsf{P}_i\mathbf{N}), \mathbf{q}_{ik})$. To make the relationship with the cost function (5) appear, we introduce a set of points $\mathbf{Q}_{ik}$ on the 3D line. Using the fact that the Euclidean distance between a point and a line is equal to the Euclidean distance between the point and the projection of this point on the line, we rewrite the line triangulation problem as:

$$\min_{\mathbf{M},\mathbf{N},\ldots,\alpha_{ik},\ldots} \mathcal{C}_{line} \text{ with } \mathcal{C}_{line} = \sum_{i=1}^{n} \sum_{k=1}^{p} d_e^2(\mathsf{P}_i(\alpha_{ik}\mathbf{M} + (1-\alpha_{ik})\mathbf{N}), \mathbf{q}_{ik}). \quad (6)$$

Compare this cost function (5): the difference is that for line triangulation, the points are not supposed to match between the different views. Hence, a 3D point on the line is reconstructed for each image point, while in the POP triangulation problem, a 3D point on the line is reconstructed for each image point correspondence. Now, the interesting point is to determine if, in practice, cost functions (5) and (6) yield close solutions for the reconstructed 3D line. Obviously, an experimental study is necessary, and we refer to §6. However, we intuitively expect that the results are close.

**Point-on-Line Triangulation.** We study the problem of point-on-line optimal triangulation: given a 3D line, represented by two 3D points $\mathbf{M}$ and $\mathbf{N}$, a set of corresponding image points $\ldots, \mathbf{q}_{ik}, \ldots$, find a 3D point $\mathbf{Q}_k \sim \alpha_k\mathbf{M} + (1-\alpha_k)\mathbf{N}$ on the given 3D line, such that the squared Euclidean distances between the predicted and the observed points is minimized.

For point-on-line triangulation, we formalise the problem as $\min_{\alpha_k} \sum_{i=1}^{n} d_e^2(\mathsf{P}_i(\alpha_k\mathbf{M} + (1-\alpha_k)\mathbf{N}), \mathbf{q}_{ik})$ and by introducing $\mathbf{b}_i = \mathsf{P}_i(\mathbf{M} - \mathbf{N})$ and $\mathbf{d}_i = \mathsf{P}_i\mathbf{N}$, we obtain:

$$\min_{\alpha_k} \mathcal{C}_{pol} \text{ with } \mathcal{C}_{pol} = \sum_{i=1}^{n} d_e^2(\alpha_k\mathbf{b}_i + \mathbf{d}_i, \mathbf{q}_{ik}). \quad (7)$$

*Sub-optimal linear algorithm.* We give a linear algorithm, based on approximating the optimal cost function (7) by replacing the Euclidean distance $d_e$ by the algebraic distance $d_a$. The algebraic cost function is $\sum_{i=1}^{n} d_a^2(\alpha_k\mathbf{b}_i + \mathbf{d}_i, \mathbf{q}_{ik}) = \sum_{i=1}^{n} \|\alpha_k\mathsf{S}[\mathbf{q}_{ik}]_\times\mathbf{b}_i + \mathsf{S}[\mathbf{q}_i]_\times\mathbf{d}_i\|^2$. A closed-form solution giving the best $\alpha_k$ in the least-squares sens is $\alpha_k = -\frac{\sum_{i=1}^{n} \mathbf{b}_i{}^\mathsf{T}[\mathbf{q}_{ik}]_\times\tilde{\mathsf{I}}[\mathbf{q}_{ik}]_\times\mathbf{d}_i}{\sum_{i=1}^{n} \mathbf{b}_i{}^\mathsf{T}[\mathbf{q}_{ik}]_\times\tilde{\mathsf{I}}[\mathbf{q}_{ik}]_\times\mathbf{b}_i}$ with $\tilde{\mathsf{I}} \sim \mathsf{S}^\mathsf{T}\mathsf{S} \sim \begin{pmatrix} 1 & & \\ & 1 & \\ & & 0 \end{pmatrix}$.

*Optimal polynomial algorithm.* This algorithm consists in finding the roots of a degree-$(3n-2)$ polynomial in the parameter $\alpha_k$, whose coefficients depend on the $\mathbf{b}_i$, the $\mathbf{d}_i$ and the $\mathbf{q}_{ik}$. Due to lack of space, details are left to an extended version of the paper.

## 5 Bundle Adjustment

Bundle adjustment consists in minimizing the reprojection error over structure and motion parameters:

$$\min_{\mathsf{P}_1,\ldots,\mathsf{P}_n,\mathbf{M}_1,\mathbf{N}_1,\ldots,\mathbf{M}_m,\mathbf{N}_m,\ldots,\alpha_{jk},\ldots} \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{p} d_e^2(\mathsf{P}_i(\alpha_{jk}\mathbf{M}_j + (1-\alpha_{jk})\mathbf{N}_j), \mathbf{q}_{ijk}),$$

where we consider without loss of generality that all points are visible in all views. We use the Levenberg-Marquardt algorithm to minimize this cost function, starting from an initial solution obtained by matching pairs of images and computing pair-wise fundamental matrices using the algorithms of §§2 and 3, from which the multiple-view geometry is extracted as in [11]. Multiple-view matches are formed, and the POPs are triangulated using the optimal method described in §4.

## 6 Experimental Results

We simulate a set of 3D POPs observed by two cameras, with focal length 1000 pixels. To simulate a realistic scenario, each POP is made of 5 supporting points. The supporting points are projected onto the images, and a Gaussian centered noise is added. The images of the supporting lines are determined as the best fit to the noisy supporting points. These data are used to compare quasi-metric reconstructions of the scene, obtained using different algorithms. We mesure the reprojection error and a 3D error, obtained as the minimum residual of $\min_{\mathsf{H}_u} \sum_j d^2(\underline{\mathbf{Q}}_j, \mathsf{H}_u\mathbf{Q}_j)$, where $\underline{\mathbf{Q}}_j$ are the groung truth 3D points, $\mathbf{Q}_j$ the reconstruction points and $\mathsf{H}_u$ an aligning 3D homography.

*Comparing triangulation algorithms.* The two first methods are based on triangulating the supporting line, then each supporting point using the linear solution (method 'Line Triangulation + Lin') or using the optimal polynomial solution (method 'Line Triangulation + Poly'). The third method is Levenberg-Marquardt minimization of the reprojection error, for POPs (method 'ML Pops') or points (method 'ML Points'). We observe on figure 3 (a) that triangulating the supporting line followed by the supporting points on this line (methods 'Line Triangulation + *') produce results close to the non-linear minimization of the reprojection error of the reprojection error of the POP (method 'ML Pops'). Minimizing the reprojection error individually for each point (method 'ML Points') produce lower reprojection errors.

Concerning the 3D error, shown on figure 3 (b), we also observe that methods 'Line Triangulation + *' produce results close to method 'ML Pop'. However,

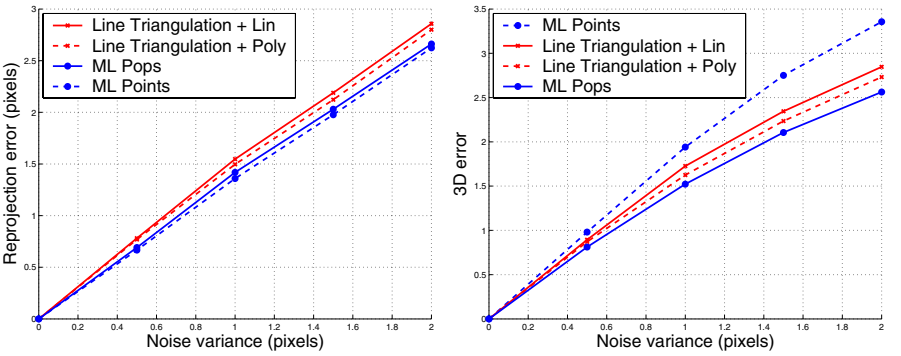



**Fig. 3.** Reprojection and 3D error when varying the added image noise variance to compare structure and motion recovery methods.

we observe that method 'ML Points' gives results worse than all other methods. This is due to the fact that this method does not benefit from the structural constraints defining POPs.

*Comparing bundle adjustment algorithms.* The two first methods are based on computing the epipolar geometry using the eight point algorithm (method 'Eight Point Alg.') or the three POP algorithm (method 'Three Pop Alg.'), then triangulating the POPs using the optimal triangulation method. The two other methods are bundle adjustment of POPs and points respectively. We observe on figure 4 (a) that the eight point algorithm yields the worse reprojection error, followed by the three POP algorithm and the eight corrected point algorithm. Bundle

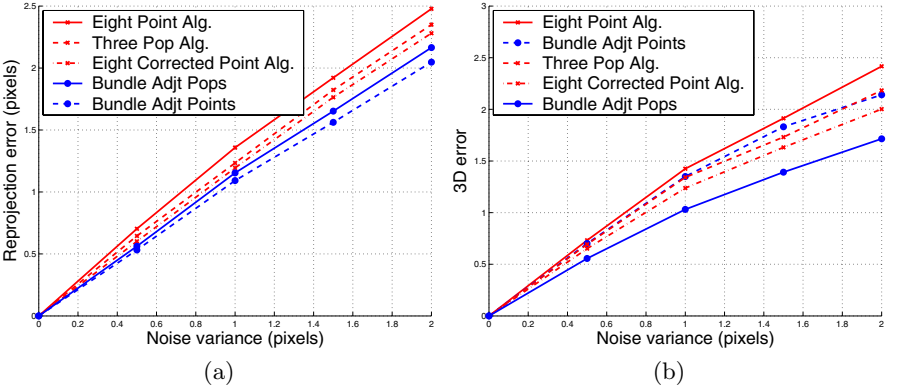



**Fig. 4.** Reprojection and 3D error when varying the added image noise variance to compare triangulation methods.

adjustement of POPs gives reprojection error slightly higher than with points. However, figure 4 (b) shows that bundle adjustment of POPs gives a better 3D structure than point, due to the structural constraints. It also shows that the eight corrected point algorithm yields good results.

## 7 Conclusions and Further Work

We addressed the problem of automatic structure and motion recovery from images containing lines. We introduced a feature that we call POP, for Pencil-of-Points. We demonstrated our matching algorithm on real images. This confirms that the repeatability rate of POPs is higher than the repeatability rates of the points and lines from which they are detected. This also shows that using POPs, wide baseline matching and the epipolar geometry can be successfully computed in an automatic manner, using simple cross-correlation. Experimental results on simulated data show that due to the strong structural constraints, POPs yield structure and motion estimates more accurate than with points.

Advantages for using POPs are numerous. Briefly, localization, repeatability rate and structure and motion estimate are better with POPs than with points, and robust estimation is very efficient since only three pairs of POPs define an epipolar geometry. For this reason, we believe that this new feature could become standard for automatic structure-and-motion in man-made environment, i.e. based on lines.

Further work will consist in investigating the determination of parameters $\boldsymbol{\mu}$ needed to compute undistorted cross-correlation, since we believe that it could strongly improve the initial matching step, and studying methods for estimating the trifocal tensor from triplets of POPs.

**Acknowledgements.** The first author would like to thank Frederik Schaffalitzky from the University of Oxford for fruitful discussions. This paper benefited from suggestions from one of the anonymous reviewers. Images of the Valbonne church have been provided by INRIA Sophia-Antipolis.

## References

1. F. Bignone, O. Henricsson, P. Fua, and M. Stricker. Automatic extraction of generic house roofs from high resolution aerial imagery. In *ECCV*, pp.85–96. April 1996.
2. J. Canny. A computational approach to edge detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.
3. M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381 – 395, June 1981.
4. R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.
5. R.I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–140, 1997.

6. R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2000.
7. H.C. Longuet-Higgins. A computer program for reconstructing a scene from two projections. *Nature*, 293:133–135, September 1981.
8. G. Médioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics and Image Processing*, 31:2–18, 1985.
9. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, volume I, pages 128–142, May 2002.
10. C. Schmid and A. Zisserman. Automatic line matching across views. In *CVPR*, pages 666–671, 1997.
11. B. Triggs. Linear projective reconstruction from matching tensors. *Image and Vision Computing*, 15(8):617–625, August 1997.
12. Z. Zhang. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27(2):161–195, March 1998.

# What Do Four Points in Two Calibrated Images Tell Us about the Epipoles?*

David Nistér[1] and Frederik Schaffalitzky[2]

[1] Sarnoff Corporation
CN5300, Princeton, NJ 08530, USA
dnister@sarnoff.com
[2] Australian National University
fsm@robots.ox.ac.uk

**Abstract.** Suppose that two perspective views of four world points are given, that the intrinsic parameters are known, but the camera poses and the world point positions are not. We prove that the epipole in each view is then constrained to lie on a curve of degree ten. We give the equation for the curve and establish many of the curve's properties. For example, we show that the curve has four branches through each of the image points and that it has four additional points on each conic of the pencil of conics through the four image points. We show how to compute the four curve points on each conic in closed form. We show that orientation constraints allow only parts of the curve and find that there are impossible configurations of four corresponding point pairs. We give a novel algorithm that solves for the essential matrix given three corresponding points and one epipole. We then use the theory to describe a solution, using a 1-parameter search, to the notoriously difficult problem of solving for the pose of three views given four corresponding points.

## 1 Introduction

Solving for unknown camera locations and scene structure given multiple views of a scene has been a central task in computer vision for several decades and in photogrammetry for almost two centuries. If the intrinsic parameters (such as focal lengths) of the views are known a priori, the views are said to be calibrated. In the calibrated case, it is possible to determine the relative pose between two views up to ten solutions and an unknown scale given five corresponding points [8,1]. In the uncalibrated case, at least seven corresponding points are required to obtain up to three solutions for the fundamental matrix, which is the uncalibrated equivalent to relative pose [4]. We will characterise the solutions in terms of their epipoles, i.e. the image in one view of the perspective center of the other view. If we have one point correspondence less than the minimum required, we can expect to get a whole continuum of solutions. In the uncalibrated case,

* Prepared through collaborative participation in the Robotics Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0012. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 41–57, 2004.
© Springer-Verlag Berlin Heidelberg 2004

it is well known [7] that six point correspondences give rise to a cubic curve of possible epipoles. However, to the best of our knowledge, the case of four point correspondences between two calibrated views has not been studied previously. We will show that four point correspondences beween two calibrated views constrain the epipole in each image to lie on a decic (i.e. tenth degree) curve. Moreover, if we disregard orientation constraints, each point on the decic curve is a possible epipole. The decic curve varies with the configuration of the points and cameras and can take on a wide variety of beautiful and intriguing shapes. Some examples are shown in Figure 1.

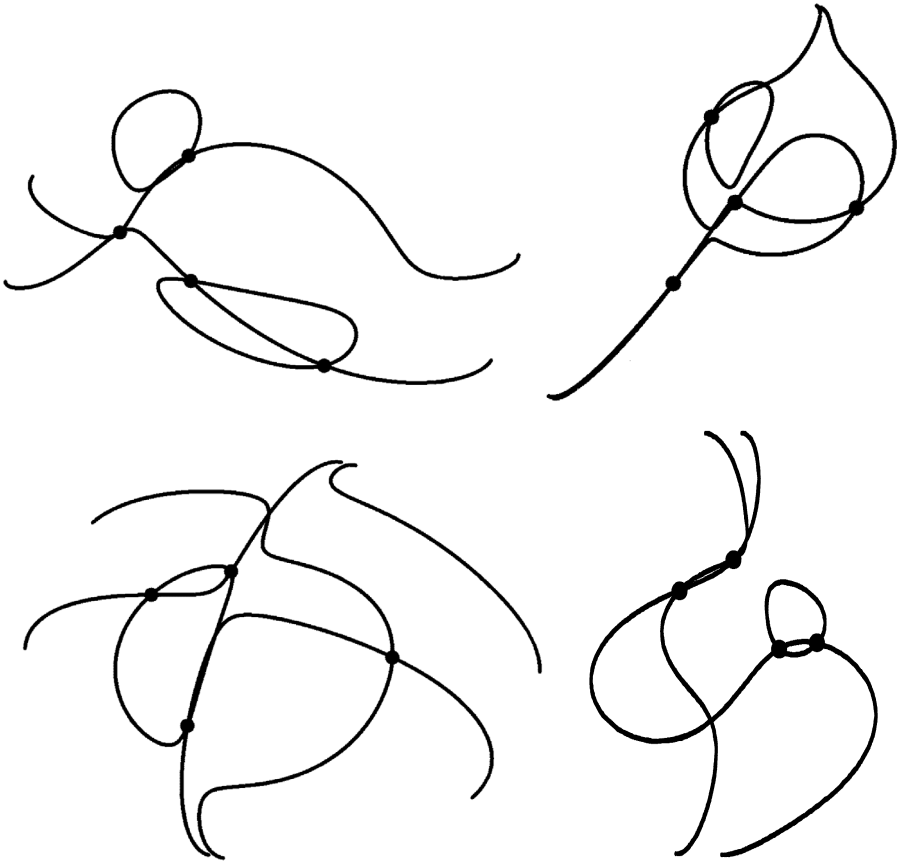


**Fig. 1.** Some examples of decic curves of possible epipoles given four points in two calibrated images.

Finally we apply the theory to describe a solution to the 3 view 4 point perspective pose problem (3v4p problem for short), which amounts to finding the relative poses

of three calibrated perspective views given 4 corresponding point triplets. The 3v4p problem is notoriously difficult to solve, but has a unique solution in general [5,10]. It is in fact overconstrained by one, meaning that four random point triplets in general can not be realised as the three calibrated images of four common world points. Adjustment methods typically fail to solve the 3v4p problem and no practical numerical solution is known. Our theory leads to an efficient solution to the 3v4p problem that is based on a one-dimensional exhaustive search. The search procedure evaluates the points on the decic curve arising from two of the three views. Each point can be evaluated and checked for three view consistency with closed form calculations. The solution minimises an image based error concentrated to one point. This is reminiscent of [12], which works with three or more views in the uncalibrated setting, see also [11]. Our algorithm can also be used to determine if four image point triplets are realisable as the three calibrated images of four common world points.

Many more point correspondences than the minimal number are needed to obtain robust and accurate solutions for structure and motion. The intended use for our 3v4p solution is as a hypothesis generator in a hypothesise-and-test architecture such as for example [8,9,2]. Many samples of four corresponding point triplets are taken and the solutions are scored based on their support among the whole set of observations.

The rest of the paper is organised as follows. In Section 2, we establish some notation and highlight some known results. In Section 3, we describe the geometric construction that serves as the basis for the main discoveries of the paper. In Sections 4 and 5, we work out the consequences of the geometric construction. In Section 6, we give the algebraic expression for the decic curve. In Section 7 we establish further properties of the curve of possible epipoles and in Section 8 we reach our main result. In Section 9 we investigate implications of orientation constraints. In Section 10 the 3v4p algorithm is given. Section 11 concludes.

## 2 Preliminaries

We broadly follow the notational conventions in [4,13]. Image points are represented by homogeneous 3-vectors $x$. Plane conics are represented by $3 \times 3$ symmetric matrices and we often refer to such a matrix as a conic. The symbol $\sim$ denotes equality up to scale. We use the notation $A^*$ to denote the adjugate matrix of $A$, namely, the transpose of the cofactor matrix of $A$. We will use $|A|$ to denote the determinant and $tr(A)$ to denote the trace of the matrix $A$. We assume the reader has some background in multi-view geometry and is familiar with concepts such as camera matrices, the absolute conic, the image of the absolute conic (IAC) under a camera projection and its dual (the DIAC). When discussing more than one view, we generally use prime notation to indicate quantities that are related to the second image; for example $x$ and $x'$ might be corresponding image points in the first and second view, respectively. Similarly, we use $e$ and $e'$ to denote the two epipoles and $\omega$ and $\omega'$ to denote the IACs in two views.

Given corresponding image points $x_i \leftrightarrow x'_i$ in two views the *epipolar constraint* [7, 1] is:

**Theorem 1.** *The projective parameters of the rays between $e$ and $x_i$ are homographically related to the projective parameters of the rays between $e'$ and $x'_i$.*

The situation is illustrated in Figure 2. The condition asserts the existence of a 1D homography that relates the pencil of lines through $e$ to the pencil of lines through $e'$. This homography is called the *epipolar line homography*.

The epipolar constraint relates corresponding image points. For the pair $\omega$, $\omega'$ of corresponding *conics* we have the *Kruppa constraints*:

**Theorem 2.** *The two tangents from $e$ to the IAC $\omega$ are related by the epipolar line homography to the two tangents from $e'$ to the IAC $\omega'$.*

The constraint is illustrated in Figure 2.

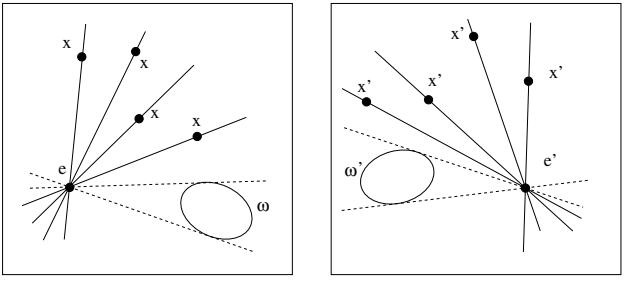



**Fig. 2.** Illustration of Theorems 1 and 2. The diagram shows two images, each with four image points, a conic and an epipole. The pencils of rays from the epipoles to the image points are related by the epipolar line homography. Similarly, the epipolar tangents to the images of the absolute conics are also related by the epipolar line homography.

These algebraic constraints treat the pre-image of an image point as an infinite line extending both backwards and forwards from the projection centre. However, the image rays are in reality half-lines extending only in the forward direction. Moreover, unless our images have been mirrored, we typically know which direction is forward. The constraint that any observed world point should be on the forward part is referred to as the orientation constraint. The orientation constraints imply that the epipolar line homography is oriented and thus preserves the orientation of the rays in Theorem 1, see [14] for more details.

## 3 The Geometric Construction

Assume that we have two perspective views of four common but unknown world points and that the intrinsic parameters of the cameras are known but their poses are not. Let the image correspondences be $x_i \leftrightarrow x'_i$. In general, no three out of the four image points

in either image are collinear and we shall henceforth exclude the collinear case from further consideration.

Accordingly, we may then choose [13] projective coordinates in each image such that the four image points have the same coordinates in both images. In other words, we may assume that $x_i = x_i'$ and we shall do this henceforth and think of both image planes as co-registered into one coordinate system.

It then follows from Steiner's and Chasles's Theorems [13] that the constraint from Theorem 1 can be converted into:

**Theorem 3.** *The epipoles $e$, $e'$ and the four image points must lie on a conic $B$. Conversely, two epipoles $e$, $e'$ that are conconic with the four image points can satisfy the epipolar constraint.*

An illustration is given in Figure 3. This conic will be important in what follows and the reader should take note of it now. When $e$ and $e'$ are conconic with the four image points, there is a *unique* epipolar line homography that makes the four lines through $e$ correspond to the four lines through $e'$. One way to appreciate $B$ is to note that we can parameterize the pencil of lines through $e$ (or $e'$) by the points of $B$ and that *corresponding lines of the two pencils meet the conic $B$ in the same point*. Armed with this observation we can translate the Kruppa constraints into:

**Theorem 4.** *The Kruppa constraints are equivalent to the condition that the two tangents to $\omega$ from $e$ intersect $B$ in the same two additional points as the two tangents to $\omega'$ from $e'$.*

This geometric construction will serve as a foundation for the rest of our development. The situation is depicted in Figure 3. Loosely speaking, the two projections (from the epipoles) of the IACs onto $B$ must coincide.

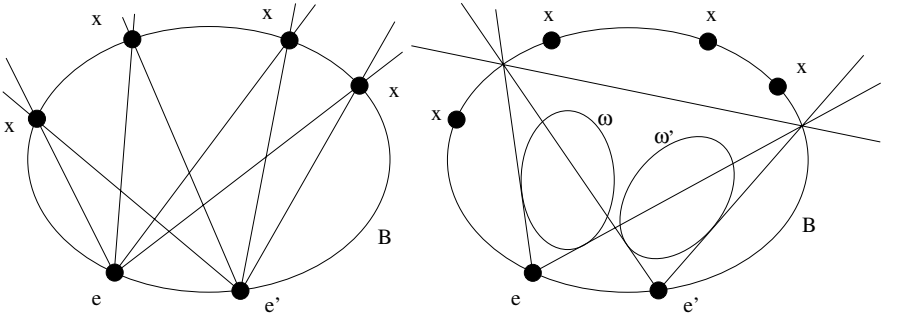



**Fig. 3.** Left: Illustration of Theorem 3. When the two image planes are co-registered so that corresponding image points coincide, the epipoles are conconic with the four image points. Right: The geometric construction corresponding to Theorem 4. The images of the IACs $\omega$,$\omega'$ made by projecting through the epipoles and onto $B$ have to coincide. This construction is the basis for the rest of our development.

## 4 Projection onto the Conic $B$

To make progress from Theorem 4 we will work out how to perform the projection of an IAC $\omega$ onto a conic $B = B(e)$ that is determined by an epipole and the four image points. One can think of the projection as being defined by the two points where the tangents to the IAC from an epipole meet $B$. But the two tangents do not come in any particular order, which is a nuisance. To avoid this complication we use the line joining the two intersection points on $B$ as our representation. This is accomplished by:

**Theorem 5.** *The projection of the IAC $\omega$ onto the proper conic $B$ through the epipole $e$ is given by the intersections of the line $(\omega \diamond B)e$ with $B$, where we define the conic*

$$(\omega \diamond B) \equiv 2B\omega^* B - tr(\omega^* B)B. \tag{1}$$

**Proof:** [1] We may choose [13] the coordinate system such that $B$ is parameterized by $\begin{bmatrix}\theta^2 & \theta & 1\end{bmatrix}^\top$ where $\theta$ is scalar. Let $\theta$ correspond to $e$ and let $\lambda$ parameterise an additional point on $B$. The line through the two points defined by $\theta$ and $\lambda$ is then $l(\theta, \lambda) = \begin{bmatrix}1 & -(\theta+\lambda) & \theta\lambda\end{bmatrix}^\top$. This line is tangent to $\omega$ when $l^\top \omega^* l = 0$, which by expanding both expressions can be seen equivalent to $\begin{bmatrix}\lambda^2 & \lambda & 1\end{bmatrix} (\omega \diamond B) \begin{bmatrix}\theta^2 & \theta & 1\end{bmatrix}^\top = 0$. Hence, the projection of $\omega$ onto $B$ through the point $\begin{bmatrix}\theta^2 & \theta & 1\end{bmatrix}^\top$ is defined by the intersections of the line $(\omega \diamond B) \begin{bmatrix}\theta^2 & \theta & 1\end{bmatrix}^\top$ with $B$. The symmetric matrix $(\omega \diamond B)$ thus represents a conic locus that has the properties stated in the theorem. Using the properties of trace, it can be verified that Equation (1) is a projectively invariant formula for a conic. The theorem follows. □

The situation is illustrated in Figure 4. We combine Theorem 4 and 5 to arrive at

**Theorem 6.** *Given that $e$ and $e'$ are conconic with the four image points, the Kruppa constraints are equivalent to the constraint that the polar lines $(\omega \diamond B)e$ and $(\omega' \diamond B)e'$ coincide.*

## 5 The Four Solutions

Note that $(\omega \diamond B) = B(\omega \cdot B)$ where we define the homography

$$(\omega \cdot B) \equiv 2\omega^* B - tr(\omega^* B)I. \tag{2}$$

In view of this, we can "cancel" a $B$ in Theorem 6 and arrive at:

[1] This theorem is a stronger version of Theorems 2 and 3, pages 179-180 in [13], which do not give a formula for $(\omega \diamond B)$. It is possible, but not necessary for our purposes, to describe $(\omega \diamond B)$ in classical terminology by saying that $\omega$ is the harmonic envelope of $B$ and $(\omega \diamond B)$. A given $\omega$ defines a correspondence $B \leftrightarrow (\omega \diamond B)$ between plane conics in the sense that $\omega \diamond (\omega \diamond B) \sim B$. Note also that the $\diamond$ operator is not commutative: projecting $\omega$ onto $B$ is different from projecting $B$ onto $\omega$.

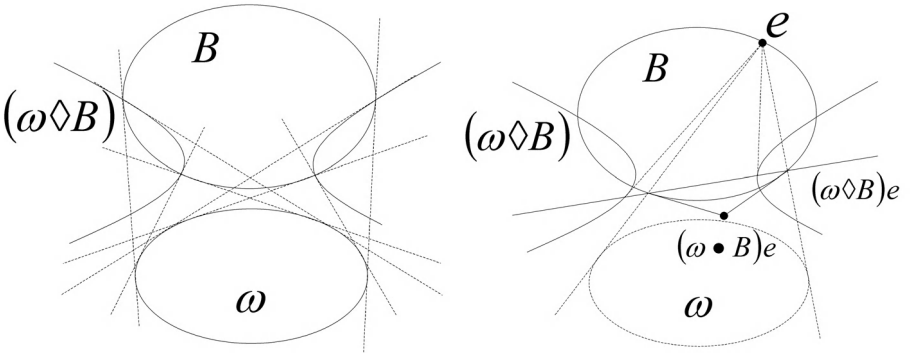



**Fig. 4.** The conic locus $(\omega \diamond B)$ from Theorem 5. Note that the line $(\omega \diamond B)e$ is the polar line of $e$ with respect to $(\omega \diamond B)$. This means that we are using the pole-polar relationship defined by the conic locus $(\omega \diamond B)$ to perform the projection. Equation (1) shows that $(\omega \diamond B)$ belongs to the pencil of conics determined by $B$ and $B\omega^* B$. This is a manifestation of the fact that $(\omega \diamond B)$ goes through the four points where the double tangents between $\omega$ and $B$ touches $B$. These four points lie on both $B$ and $B\omega^* B$. Moreover, the double tangents between $\omega$ and $(\omega \diamond B)$ touches $(\omega \diamond B)$ at the same four points. On the right, the line $(\omega \diamond B)e$ and its pole $(\omega \cdot B)e$ (see the following section for the definition of $(\omega \cdot B)$) with respect to $B$ is shown. Both can be used to represent the projection of $\omega$ onto $B$ through $e$.

**Theorem 7.** *The epipoles are related by the seventh degree mapping*

$$e' \sim (\omega' \cdot B)^*(\omega \cdot B)e. \tag{3}$$

*The Kruppa constraints single out four solutions* [2] *for the epipole* $e$ *on each proper conic* $B$*. The solutions are the intersections of* $B$ *with the conic*

$$C \equiv (\omega \cdot B)^\top (\omega' \cdot B)^{*\top} B(\omega' \cdot B)^*(\omega \cdot B). \tag{4}$$

*On the three conics* $B$ *of the pencil for which* $|(\omega' \cdot B)| = 0$*, the four solutions group into two pairs of coincident solutions.*

**Proof:** Any two conics $B_1, B_2$ from the pencil can be used as a basis for the pencil and we have

$$B(e) = (e^\top B_2 e)B_1 - (e^\top B_1 e)B_2, \tag{5}$$

i.e. $B(e)$ can be expressed quadratically in terms of $e$. According to Theorem 6, $(\omega' \diamond B)e' \sim (\omega \diamond B)e$, which for proper $B$ is equivalent to $(\omega' \cdot B)e' \sim (\omega \cdot B)e$. If we assume that $|(\omega' \cdot B)| \neq 0$, this is equivalent to Equation (3), which is seen to be a 7-th degree mapping. Since $e'$ has to be on $B$, i.e. $e'^\top B e' = 0$, we get that $e$

[2] Apart from the four image points.

must be on the conic $C$. The rest of the theorem follows from detailed consideration of the case when $(\omega' \cdot B)$ becomes rank 2, in which case $C$ degenerates to a repeated line. □

If we use Equation (5) to express $B$ in terms of $e$, then $C$ is a 14-th degree function of $e$ and we get

**Theorem 8.** *An epipole hypothesis $e$ that gives rise to a proper conic $B$ satisfies the 16-th degree equation*

$$e^\top C e = 0 \tag{6}$$

*if and only if it can satisfy the epipolar and Kruppa constraints.*

An example plot of the set of points that satisfy Equation (6) is shown in Figure 5. As indicated by the following theorem, it also includes the six lines through all pairs of image points as factors:

**Theorem 9.** *For degenerate $B$ (consisting of a line-pair) the homography $(\omega \cdot B)$ interchanges the lines of the line-pair. As a result, the curve defined by Equation (6) contains the six lines through pairs of the four image points, i.e. it contains the factor $|B|$.*

**Proof:** As $e$ moves to approach one of the lines of a line pair, the two points defined by projecting $\omega$ onto $B$ through $e$ approaches the other line of the line pair. Hence, the line $(\omega \diamond B)e$ becomes the other line. In a similar fashion, the point $(\omega \cdot B)e$, which is the pole of $(\omega \diamond B)e$ with respect to $B$, becomes a point on the other line. Thus, we get $C \sim B$ when $B$ is a line-pair. Since $e$ lies on $B$ per definition of $B$, the theorem follows. □

Hence, Equation (6) defines a superset of the possible epipoles as determined by the epipolar and Kruppa constraints. However, not all the points on the six lines are allowed by the Kruppa constraints. In fact, since Theorem 4 applies for any $B$, one can work out the consequences of the geometric construction specifically for degenerate $B$ in a similar fashion as for proper $B$. This leads to the following theorem, which we state without proof:

**Theorem 10.** *Apart from the four image points, there are at most two possible epipoles $e$ on any line joining a pair of the four image points.*

## 6 The Decic Expression

The algebraic endeavour of eliminating the factor $|B|$ from Equation (6) to arrive at a decic expression is surprisingly involved. We will just state the result. Define

$$D \equiv \omega^*, \qquad t \equiv tr(DB), \qquad U \equiv (\omega \cdot B) = 2DB - tI \tag{7}$$

and analogously for the primed entities. Then the decic expression is

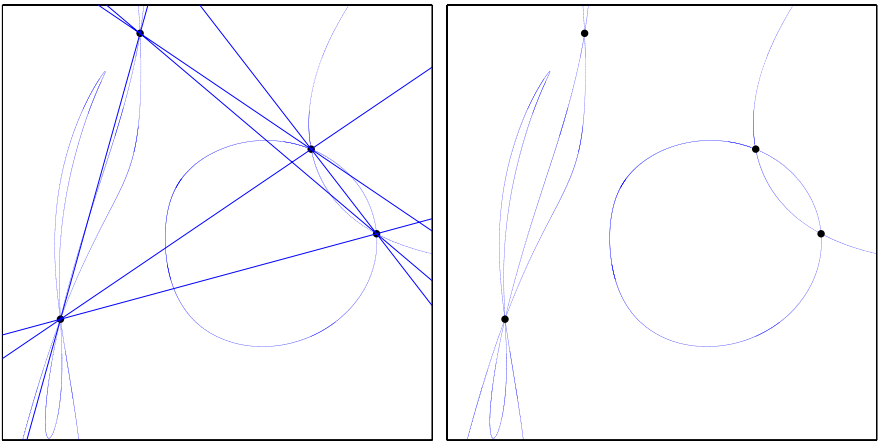


**Fig. 5.** The 16-th degree expression in Equation (6) defines a superset of the possible epipoles as determined by the epipolar and Kruppa constraints. However, it also includes the six lines through all pairs of image points as factors, which can also be eliminated. The plot on the left shows the 16-th degree curve, including the six lines, and the plot on the right shows the decic curve resulting when removing the six lines. The four small black dots are image points.

$$e^\top G(e) e = 0, \tag{8}$$

where G(e) is the conic defined by the symmetric part of

$$4U^\top D'^* B^* D'^* U + 2t' U^\top D'^* U' U + 4t'^2 BDD'^* (DB - tI) - 4t'^2 tr(B^* D'^*) D^* + t'^4 D^* + t^2 t'^2 D'^*. \tag{9}$$

$G(e)$ can readily be seen to be octic in terms of $e$, since $D$ is constant and $B$, $U$ and $t$ are all quadratic in $e$. Some examples of the decic are shown in Figure 1.

## 7 Further Properties of the Curve

The decic expression (8) is in fact exactly the set of possible epipoles under the epipolar and Kruppa constraints. The following leads to a property of the set of possible epipoles that serves as a cornerstone in getting this result.

**Theorem 11.** *Given three point correspondences $x_i \leftrightarrow x'_i$ and the epipole $e$ in one image, the epipolar and Kruppa constraints lead to four solutions for the other epipole $e'$.*

**Proof:** To prove this we give a novel constructive algorithm in Appendix A.

**Theorem 12.** *The set of possible epipoles according to the epipolar and Kruppa constraints has exactly four branches through each of the four image points.* [3] *The branches are continuous.* [4] *For points in general position, 0,2 or 4 of the branches can be real.*

[3] We allow the world points to coincide with the projection centres of the cameras.

[4] Assume the joint epipole $(e, e')$ describes a smooth curve in $\mathbb{P}^2 \times \mathbb{P}^2$. When we talk about tracing a curve branch in an image we really have in mind tracing the curve in $\mathbb{P}^2 \times \mathbb{P}^2$. The

**Proof:** According to Theorem 11 we have that for a general epipole position $e$, there are four solutions for the essential matrix that obey three given point pairs. When $e$ coincides with one of the image points, $x_1$ say, all four solutions that obey the other three image point pairs also satisify $x_1 \leftrightarrow x_1'$, since the epipolar line $l'$ joining $e'$ and $x_1'$ always map into a line $l$ through $x_1$. Moreover, the line $l$ determined by the other three image point pairs changes continuously if we change $e$ and hence $l$ has to be the tangent direction of the corresponding curve branch. It is in fact the same as the tangent at $x_1$ to the conic determined by the four image points and the corresponding solution for $e'$. The tangent direction of each branch can thus be computed in closed form with the algorithm in Appendix A. It is clear from the algorithm that points *in general position* can not give rise to an odd number of real solutions. Using the algorithm, we have found examples of cases with 0, 2 and 4 real solutions. □

The situation is illustrated in Figure 6. Two real branches is by far most common.

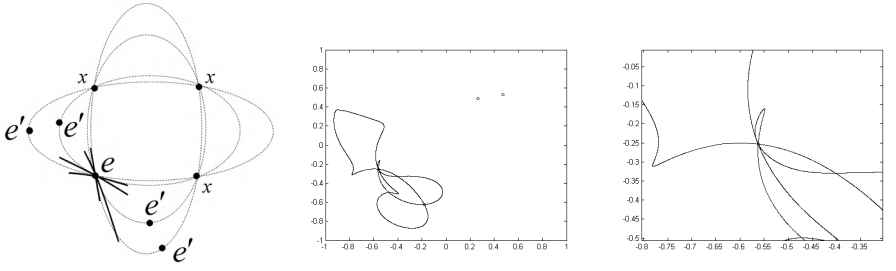



**Fig. 6.** Left: As indicated by Theorem 12, there are four branches of the curve of possible epipoles through each of the four image points. Each curve branch is tangent to the conic $B$ that includes the four image points and the epipole $e'$ corresponding to $e$ coincident with the image point according to Theorem 12 and Appendix A. Middle: An example of a decic curve with 0, 2 and 4 real branches through the image points. The image points are marked with small circles. Right: Close-up.

## 8 Main Result

**Theorem 13.** *An epipole hypothesis $e$ satisfies the epipolar and Kruppa constraints if and only if it lies on the decic curve defined by Equation (8).*

**Proof:** We give a sketch of the detailed proof, which takes several pages. It follows from Theorems 8 and 9 that the possible epipoles off of the six lines are described by the decic.

curve projects into $\mathbb{P}^2$ in such a way that four points map to each image point $x_i$ and the four non-intersecting branches in $\mathbb{P}^2 \times \mathbb{P}^2$ project to the four intersecting branches of the image curve.

The key is then to use Theorems 7 and 12 to establish that ten degrees are necessary and that the decic does not have any redundant factors, which follows from Bezout's Theorem when considering the number of possible $e$ on general $B$. Finally, by Theorem 10 and continuity of the geometric construction, one can establish that the decic intersects any one of the six lines at the correct two additional points apart from the image points. □

We can also get a more complete version of Theorem 7 that applies even when the conic $B$ from Equation (5) is degenerate.

**Theorem 14.** *Given any point $y$ distinct from the four image points, the possible epipoles on the conic $B(y)$ according to the epipolar and Kruppa constraints are exactly the four image points plus the intersections between the two conics $B(y)$ and $G(y)$.*

**Proof:** According to Theorem 13, a point $e$ is a possible epipole iff it lies on $G(e)$. An epipole hypothesis $e$ apart from the four image points generates the same conic $B(e) = B(y)$ as $y$ iff it lies on $B(y)$. Since $G(e)$ can be written as a function of $B$ only, all $e$ on $B(y)$ apart from the four image points generate the same $G(e) = G(y)$. Thus, the points on $B(y)$ apart from the four image points satisfy the decic iff they lie on $G(y)$. Finally, the four image points lie on $B(y)$ by construction and they are always possible epipoles. □

**Theorem 15.** *The curve of possible epipoles according to the epipolar and Kruppa constraints has exactly ten singular points. The four image points each have multiplicity four. In addition, there are exactly three pairs of nodal points with multiplicity two.* [5] *The three pairs of nodal points occur on the three conics $B$ for which $|(\omega' \cdot B)| = 0$. These conics are exactly the three $B$ of the pencil with an inscribed quadrangle that is also circumscribed to $\omega'$.*

**Proof:** Recall Theorem 6 and observe that on proper conics $B$, the solution $e$ has multiplicity two exactly when the line $l = (\omega \diamond B)e$ obtained by projecting $\omega$ onto $B$ through $e$ also can be obtained by projecting $\omega'$ onto $B$ in two distinct ways through two distinct points $e'$ on $B$. This is illustrated in Figure 7. According to Poncelet's Porism [13], given proper conics $B$ and $\omega'$, we have two possibilities. Either there is no quadrangle inscribed in $B$ that is also circumscribed to $\omega'$, or there is one such quadrangle with any point on $B$ as one of its vertices. In the former case, no epipole hypothesis $e'$ on $B$ in the second image ever generates the same line $l' = (\omega' \diamond B)e'$ as some other epipole hypothesis. Hence no solution for $e$ can then have multiplicity two. In the latter case, every epipole hypothesis $e'$ generates the same line $l'$ as exactly one other epipole hypothesis. Thus, in this case the solutions $e$ on $B$ always have multiplicity two. The latter case has to happen exactly when $|(\omega' \cdot B)| = 0$ and we see that this must be the same as the condition that there is a quadrangle inscribed in $B$ that is also circumscribed to $\omega'$. The remaining parts of the theorem follow from Theorems 7, 12 and 13. □

[5] By the degree-genus formula [6] the genus of the curve is therefore $((10-1)(10-2) - 4 \times 4(4-1) - 6 \times 2(2-1))/2 = 6$ so, in particular, the curve is not rational.

## 9 Bringing in the Orientation Constraint

The orientation constraint asserts that the space point corresponding to a visible image point must lie in front of the camera. The situation is illustrated in Figure 7. Given $e$ on the decic, verification of the orientation constraints is straightforward. Equation (3) determines $e'$. The epipolar line homography, and hence the essential matrix, is then determined by the point correspondences. It is well known [8] that the essential matrix corresponds to four possible 3D configurations and that the orientation information from a corresponding point pair singles out one of them. The orientation constraints can be satisfied exactly when all four point correspondences indicate the same configuration. We will split the orientation constraint into two conditions:

Firstly, the two forward half-rays of an image correspondence lie in the same half-plane (the baseline separates each epipolar plane into two half-planes). Then the common space point is either on the forward part of both rays, or on the backward part of both rays. This condition is called the *oriented epipolar constraint* because it is satisfied exactly when the epipolar line homography is oriented [14].

Secondly, the forward half-rays should converge in their common half-plane. We will refer to this as the *convergence constraint*.

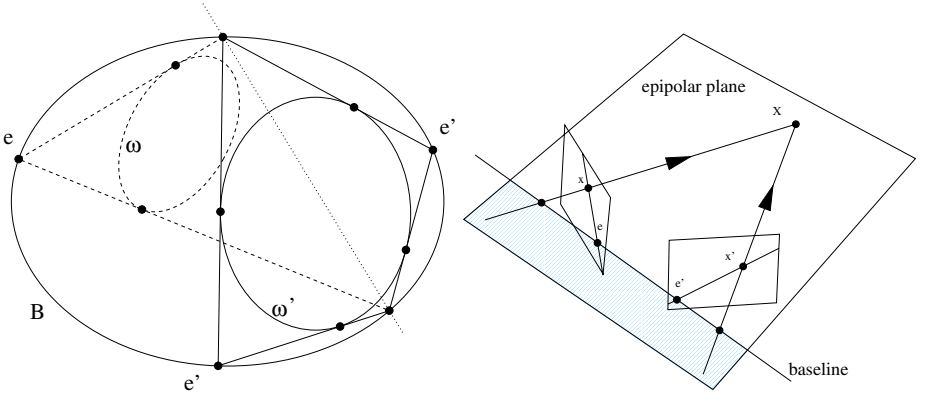



**Fig. 7.** Left: For there to be multiple $e'$ corresponding to one $e$, there has to be a quadrangle inscribed in $B$ that is also circumscribed to $\omega'$. According to Poncelet's porism, there is either no such quadrangle, or a whole family of them. Right: The orientation constraint is that space points should be in the forward direction on their respective image rays. It can be partitioned into requiring 1) that the forward half-rays point into the same half-plane and 2) that the half-rays converge in that half-plane.

**Theorem 16.** *The satisfiability of the oriented epipolar constraint can only change at those points $e$ of the decic curve for which $e$ or one of its possible corresponding $e'$ coincide with one of the four image points, i.e. only at the four image points or at the*

*up to* $4 \times 4$ *real points that correspond to one of the four image points according to Theorem 12 and the algorithm in Appendix A.*

**Proof:** When we move $e$ along the decic curve and neither $e$ nor its corresponding $e'$ coincides with one of the four image points, $e'$ and the epipolar line homography changes continuously with $e$. Moreover, since the epipolar constraint is satisfied for all points on the decic, the ray orientations can not change unless one of the epipoles coincides with one of the four image points. □

The situation is illustrated in Figure 8.

**Theorem 17.** *A branch of the decic through an image point can have at most one side allowed by the oriented epipolar constraint. One side is allowed iff the epipolar line homography accompanying the pair of epipoles* $e, e'$ *corresponding to the branch is oriented with respect to the other three image correspondences.*

**Proof:** For a particular branch, through $x$ say, $e'$ and the epipolar line homography changes continuously with $e$, so the orientations of the rays to the other three image points do not change at $x$. However, the orientation of the ray to $x$ changes when $e$ passes through $x$ and the theorem follows. □

For points on the decic, there is a valid epipolar geometry. The rays can only change between convergent and divergent when they become parallel. This can only occur when the angles between the image points and the epipoles are equal in both images. The Euclidean scalar product between image directions $x$ and $y$ is encoded up to scale in the IAC $\omega$ as $<x|y>= x^\top \omega y$. Using this and $e' \sim U'^* U e$ it can be shown that parallelism can only occur when [6]

$$(x^\top \omega' x)(e^\top U^\top U'^{*\top} \omega' U'^* U e)(x^\top \omega e)^2 - (x^\top \omega x)(e^\top \omega e)(x^\top \omega' U'^* U e)^2 = 0, \tag{10}$$

which is a 16th degree expression in $e$. Its intersections with the decic are the only places where the rays from $x$ can change between converging and diverging.

## 10 The 3v4p Algorithm

Given four point correspondences in three calibrated views, we choose two views and trace out the decic curve for those two views with a one-dimensional sweep driven by a parameter $\theta$. For each value of $\theta$, all computations can be carried out very efficiently in closed form. The parameter is used to indicate one conic $B$ from the pencil of conics. Given $B$, we calculate the conic $G$ from Equation (9) as a function of $B$ and the intersections between $G$ and $B$ can then be found in closed form as the roots of a quartic polynomial. This yields up to four solutions for $e$. For each solution, the corresponding $e'$ can be found through Equation (3). If we rotate both coordinate systems so that the

[6] Remember that we are assuming that the points are co-registered.

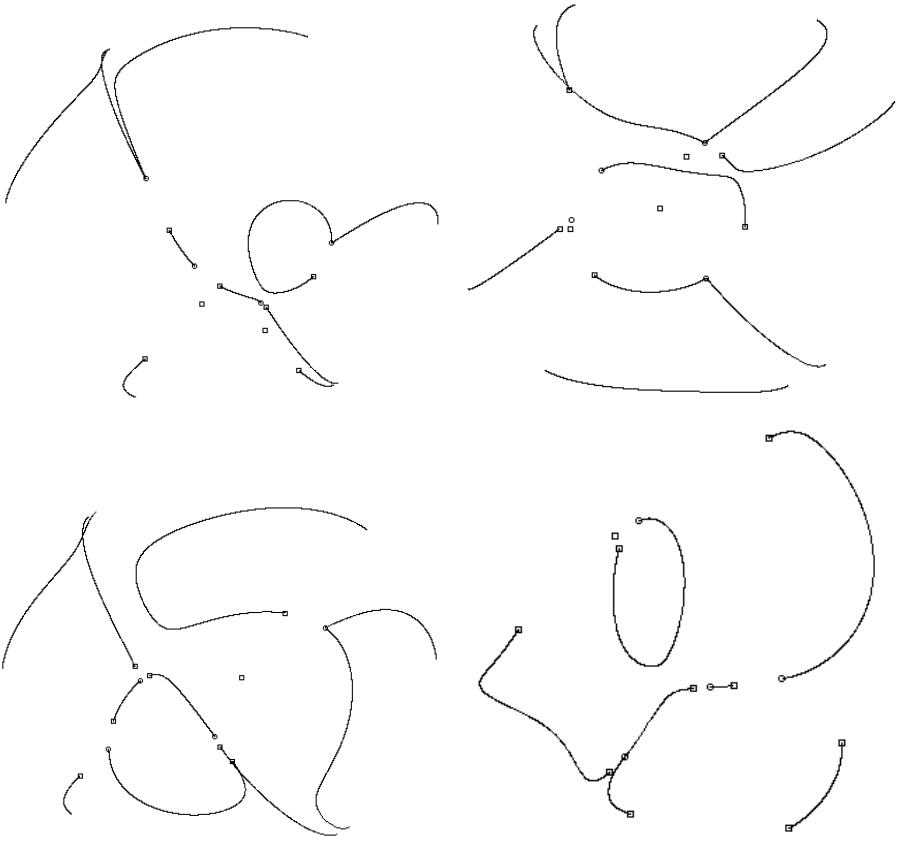


**Fig. 8.** Examples of curves of possible epipoles after the oriented epipolar constraint and the Kruppa constraints have been enforced. The small circles mark the four image points, while the squares mark the up to $4 \times 4$ real points that correspond to one of the four image points according to Theorem 12 and the algorithm in Appendix A. As indicated by Theorem 16, the curve has no loose ends apart from those points. The curves are rendered as the orthographic projection of a half-sphere and all curve segments that appear loose actually reappear at the antipode. There are also configurations of four point pairs for which the set of possible epipoles is completely empty, i.e. all configurations of four points in two calibrated views are not possible according to the oriented epipolar constraint.

epipoles are moved to the origin, finding the epipolar line homography is just a simple matter of solving for a 1-D rotation with possible reflection. Thus, we get the essential matrix for the two views corresponding to each solution. Following [8], we can then select a camera configuration for the two views and get the locations of the four points through triangulation. Each solution then leads to up to four solutions for the pose of the third view when solving the three point perspective pose problem [3] for three of the points. The orientation constraints are used to disqualify solutions for which the space

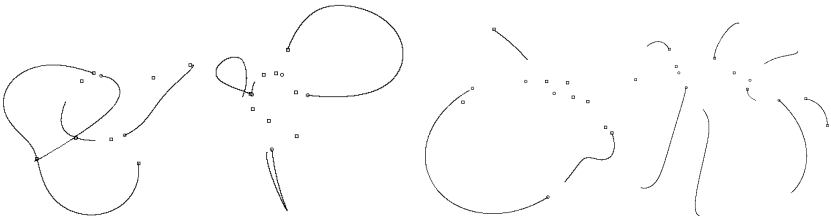


**Fig. 9.** Possible epipoles when all constraints (epipolar, Kruppa and full 3D orientation) are enforced.

points are not on the forward part of the image rays. Finally, the fourth point can be projected into the third view. For the correct value of $\theta$ and the correct solution, the projection of the fourth point should coincide with its observed image position. Moreover, this will only occur for valid solutions and in general there is a unique solution. Thus, $\theta$ is swept through the pencil of conics and the solution resulting in the reprojection closest to the observed fourth point position is selected.

This algorithm has been implemented and shown to be very effective in practice. Experimental results will appear in an upcoming journal paper.

## 11 Conclusion

We have given necessary and sufficient conditions for the epipolar and Kruppa constraints to be satisfied given four corresponding points in two calibrated images. The possible epipoles are exactly those on a decic curve. We have shown that the second epipole is related to the first by a seventh degree expression. We have shown that if the orientation constraints are taken into account, only a subset of the decic curve corresponds to possible epipoles. As a result, we have found that there are configurations of four pairs of corresponding points that can not occur in two calibrated images. This is similar in spirit to [14]. We have shown that points on the decic curve can be generated in closed form and that it is possible to trace out the curve efficiently with a one-dimensional sweep. This yields a solution to the notoriously difficult problem of solving for the relative orientation of three calibrated views given four corresponding points. In passing, we have given a novel algorithm for finding the essential matrix given three point correspondences and one of the epipoles.

## References

1. O. Faugeras, *Three-Dimensional Computer Vision: a Geometric Viewpoint*, MIT Press, ISBN 0-262-06158-9, 1993.
2. M. Fischler and R. Bolles, Random Sample Consensus: a Paradigm for Model Fitting with Application to Image Analysis and Automated Cartography, *Commun. Assoc. Comp. Mach.*, 24:381-395, 1981.

3. R. Haralick, C. Lee, K. Ottenberg and M. Nölle, Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem, *International Journal of Computer Vision*, 13(3):331-356, 1994.
4. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN 0-521-62304-9, 2000.
5. R. Holt and A. Netravali, Uniqueness of Solutions to Three Perspective Views of Four Points, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 17, No 3, March 1995.
6. F. Kirwan, *Complex Algebraic Curves*, Cambridge University Press, ISBN 0-521-42353-8, 1995.
7. S. Maybank, *Theory of Reconstruction from Image Motion*, Springer-Verlag, ISBN 3-540-55537-4, 1993.
8. D. Nistér. An Efficient Solution to the Five-Point Relative Pose Problem, *IEEE Conference on Computer Vision and Pattern Recognition*, Volume 2, pp. 195-202, 2003.
9. D. Nistér. Preemptive RANSAC for Live Structure and Motion Estimation, *IEEE International Conference on Computer Vision*, pp. 199-206, 2003.
10. L. Quan, B. Triggs, B. Mourrain, and A. Ameller, Uniqueness of Minimal Euclidean Reconstruction from 4 Points, *unpublished*, 2003.
11. L. Quan, Invariants of Six Points and Projective Reconstruction from Three Uncalibrated Images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(17):34-46, January, 1995.
12. F. Schaffalitzky, A. Zisserman, R. Hartley and P. Torr, A Six Point Solution for Structure and Motion, *European Conference on Computer Vision*, Volume 1, pp. 632-648, 2000.
13. J. Semple and G. Kneebone, *Algebraic Projective Geometry*, Oxford University Press, ISBN 0-19-8503636, 1952.
14. T. Werner, Constraints on Five Points in Two Images, *IEEE Conference on Computer Vision and Pattern Recognition*, Volume 2, pp. 203-208, 2003.

## A Three Points Plus Epipole

To support our arguments we need to show that given three point correspondences $x_i \leftrightarrow x_i'$ and the epipole $e$ in one image, the epipolar and Kruppa constraints lead to four solutions for the other epipole $e'$. To do this, we give a novel algorithm that constructs the four solutions. If the epipole in the first image is known, we can rotate the image coordinate system so that it is on the origin. Then the essential matrix is of the form $E = \begin{bmatrix} E_1 & E_2 & 0; & E_3 & E_4 & 0; & E_5 & E_6 & 0 \end{bmatrix}$. A point correspondence $x' \leftrightarrow x$ contributes the constraint $\tilde{X}^\top \tilde{E} = 0$, where

$$\tilde{X} = \begin{bmatrix} x_1'x_1 & x_1'x_2 & x_2'x_1 & x_2'x_2 & x_3'x_1 & x_3'x_2 \end{bmatrix}^\top \tag{11}$$

and

$$\tilde{E} = \begin{bmatrix} E_1 & E_2 & E_3 & E_4 & E_5 & E_6 \end{bmatrix}^\top. \tag{12}$$

If the vectors $\tilde{X}^\top$ from three point correspondences are stacked, we get a $3 \times 6$ matrix. $\tilde{E}$ must be in its 3-dimensional nullspace. Let $Y, Z, W$ be a basis for the nullspace. Then $\tilde{E}$ is of the form $\tilde{E} = yY + zZ + wW$, where $y, z, w$ are some scalars. Since an essential

matrix $E$ is characterised by having two equal singular values and one zero singular value, we have exactly the two additional constraints $E_1E_2 + E_3E_4 + E_5E_6 = 0$ and $E_1^2 + E_3^2 + E_5^2 = E_2^2 + E_4^2 + E_6^2$. These constraints represent two conics and four solutions for $\begin{bmatrix} y & z & w \end{bmatrix}^\top$.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

# Dynamic Visual Search Using Inner-Scene Similarity: Algorithms and Inherent Limitations

Tamar Avraham and Michael Lindenbaum

Computer Science Department
Technion, Haifa 32000, Israel
tammya,mic@cs.technion.ac.il

**Abstract.** A dynamic visual search framework based mainly on inner-scene similarity is proposed. Algorithms as well as measures quantifying the difficulty of search tasks are suggested. Given a number of candidates (e.g. sub-images), our basic hypothesis is that more visually similar candidates are more likely to have the same identity. Both deterministic and stochastic approaches, relying on this hypothesis, are used to quantify this intuition. Under the deterministic approach, we suggest a measure similar to Kolmogorov's $\epsilon$-covering that quantifies the difficulty of a search task and bounds the performance of all search algorithms. We also suggest a simple algorithm that meets this bound. Under the stochastic approach, we model the identities of the candidates as correlated random variables and characterize the task using its second order statistics. We derive a search procedure based on minimum MSE linear estimation. Simple extensions enable the algorithm to use top-down and/or bottom-up information, when available.

## 1 Introduction

Visual search is required in situations where a person or a machine views a scene with the goal of finding one or more familiar entities. The highly effective visual-search (or more generally, attention) mechanisms in the human visual system were extensively studied from psychophysics and physiology points of view. Yarbus [24] found that the eyes rest much longer on some elements of an image, while other elements may receive little or no attention. Neisser [11] suggested that the visual processing is divided into pre-attentive and attentive stages. The first consists of parallel processes that simultaneously operate on large portions of the visual field, and form the units to which attention may then be directed. The second stage consists of limited-capacity processes that focus on a smaller portion of the visual field. Triesman and Gelade (*feature integration theory* [19]) formulate an hypothesis about how the human visual system performs pre-attentive processing. They characterized (qualitatively) the difference between search tasks requiring scan (serial) and those which do not (parallel, or pop-out). While several aspects of the Feature Integration Theory were criticized, the theory was dominant in visual search research and much work was carried out based on its premises, e.g. to understand how feature integration occurs

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 58–70, 2004.
© Springer-Verlag Berlin Heidelberg 2004

(some examples are [8,23,21]). Duncan and Humphreys rejected the dichotomy of parallel vs. serial search and proposed an alternative theory based on similarity [3]. According to their theory, two types of similarities are involved in a visual search task: between the objects in the scene, and between the objects and prior knowledge. They suggest that when a scene contains several similar structural units there is no need to treat every unit individually. Thus, if all non-targets are homogeneous, they may be rejected together resulting in a fast (pop-out like) detection, while if they are heterogeneous the search is slower.

Several search mechanisms were implemented, usually in the context of HVS (human visual system) studies (e.g. [8,21,23,5]). Other implementations focused on computer vision applications (e.g. [7,17,18]), and sometimes used other sources of knowledge to direct visual search. For example, one approach is to search first for a different object, easier to detect, which is likely to appear close to the sought for target ([15,22]). Relatively little was done to quantitatively characterize the inherent difficulty of search tasks. Tsotsos [20] considers the complexity of visual search and proves, for example, that spatial boundedness of the target is essential to make the search tractable. In [22], the efficiency of indirect search is analyzed.

This work has two goals: to provide efficient search algorithms and to quantitatively characterize the inherent difficulty of search tasks. We focus on the role of inner-scene similarity. As suggested in [3], the HVS mechanism uses similarity between objects of the same identity to accelerate the search. In this paper we show that computerized visual search can also benefit from such information, while most visual search application totally ignore this source of knowledge. We take both deterministic and stochastic approaches. Under the deterministic approach, we characterize the difficulty of the search task using a metric-space cover (similar to Kolmogorov's $\epsilon$-covering [9]) and derive bounds on the performance of all search algorithms. We also propose a simple algorithm that provably meets these bounds. Under the stochastic approach, we model the identity of the candidates as a set of correlated random variables taking target/non-target values and characterize the task using its second order statistics. We propose a linear estimation based search algorithm which can handle both inner-scene similarity and top-down information, when available.

Paper outline: The context for visual search and some basic intuitive assumptions are described in Sect. 2. Sect. 3 develops bounds on the performance of search algorithms, providing measures for search tasks' difficulty. Sect. 4 describes the VSLE algorithm based on stochastic considerations In Sect. 5 we experimentally demonstrate the validity of the bounds and the algorithms' effectiveness. [1]

## 2 Framework

### 2.1 The Context – Candidate Selection and Classification

The task of looking for object/s of certain identity in a visual scene is often divided into two subtasks. One is to *select* sub-images which serve as *candidates*. The

[1] A preliminary version of the VSLE algorithm was presented in [1].

other, the *object recognition* task, is to decide whether a candidate is a sought for object or not. The candidate selection task can be performed by a segmentation process or even by a simple division of the image into small rectangles. The candidates may be of different size, bounded or unbounded [20], and can also overlap. The object recognizer is usually computationally expensive, as the object appearance may vary due to changes in shape, color, pose, illumination etc. The recognizer may need to recognize a category of objects (and not a specific model), which usually makes it even more complex.

The object recognition process gets the candidates, one by one, after some ordering. An efficient ordering, which is more likely to put the real objects first, is the key to high efficiency of the full task. This ordering is the attentional mechanism on which we focus here.

### 2.2 Sources of Information for Directing the Search

Several information sources enabling more efficient search are possible:
**Bottom-up saliency of candidates** - In modelling HVS attention, it is often claimed that a saliency measure, quantifying how every candidate is different from the other candidates in the scene, is calculated. ([19,8,7]). Saliency is important in directing attention, but it can sometimes mislead or not be applicable when, say, the scene contains several similar targets.
**Top-down approach** - When prior knowledge is available, the candidates may be ranked by their degree of consistency with the target description ([23,6]). In many cases, however, it is hard to characterize the objects of interest in a way which is effective and inexpensive to evaluate.
**Mutual similarity of candidates** - Usually, a higher inner-scene visual similarity implies a higher likelihood for similar (or equal) identity ([3]). Under this assumption, after revealing the identity of one (or a few) candidates, it can effect the likelihood of the remaining candidates to have the same/different identity.

In this paper we focus on (the less studied) mutual similarity between candidates, and assume that no other information is given. Nevertheless, we show how to handle top-down information and saliency, when available.

To quantify similarity, we embed the candidates as points in a metric space with distances reflecting dissimilarities. We shall either assume that the distance between two objects of different identities is larger than a threshold (deterministic approach), or that the identity correlation is a monotonically descending function of this distance (stochastic approach).

### 2.3 Algorithms Framework

The algorithms we propose share a common framework. They begin from an initial priority map, indicating the prior likelihood of each candidate to be a target. Iteratively, the candidate with the highest priority receives the attention. The relevant sub-image is examined by a high-level recognizer, which we denote the *recognition oracle*. Based on the *oracle*'s response and the previous priority map, a new priority map is calculated, taking into account the similarities.

Usually, systems based on bottom-up or top-down approaches suggest calculating a saliency map before the search starts, pre-specifying the scan order. This *static* map may change only to inhibit the return to already attended locations [8]. The search algorithms proposed here, however, are *dynamic* as they change the priority map based on the results of the object recognizer.

### 2.4 Measures of Performance

For quantifying the search performance, we take a simplified approach and assume that only the costs associated with calling the recognition oracle are substantial. Therefore, we measure (and predict) the number of queries required to find a target.

## 3 Deterministic Bounds of Visual Search Performance

In this section we analyze formally the difficulty of search tasks. Readers interested only in the more efficient algorithms based on a stochastic approach can skip this section and continue reading from section 4.1.

**Notations.** We consider an abstract description of a search task as a pair $(X, l)$, where $X = \{x_1, x_2, \ldots, x_n\}$ is a set of partial descriptions associated with the set of candidates, and $l : X \to \{T, D\}$ is a function assigning identity labels to the candidates. $l(x_i) = T$ if the candidate $x_i$ is a target, and $l(x_i) = D$ if $x_i$ is a non-target (or a distractor). An attention, or search algorithm, $A$, is provided with the set $X$, but not with the labels $l$. It requires $\text{cost}_1(A, X, l)$ calls to the recognizer oracle, until the first target is found. We refer to the set of partial descriptions $X = \{x_1, x_2, \ldots, x_n\}$ as points in a metric space $(S, d)$, $d : S \times S \to \mathbb{R}^+$ being the metric distance function. The partial description can be, for example, a feature vector, and the distance may be the Euclidian metric.

**A Difficulty Measure Combining Targets' Isolation and Candidates' Scattering.** We would like to develop a search task characteristic which quantifies the search task difficulty. To be effective, this characteristic should combine two main factors:

1. The feature-space-distance between target and non-target candidates.
2. The distribution of the candidates in the feature space.

Intuitively, the search is easier when the targets are more distant from non-targets. However, if the non-targets are also different from each other, the search again becomes difficult. A useful quantification for expressing a distribution of points in a metric space uses the notion of a metric cover [9].

**Definition 1.** *Let $X \subseteq S$ be a set of points in a metric space $(S, d)$. Let $2^S$ be the set of all possible subsets of $S$. $\mathcal{C} \subset 2^S$ is 'a cover' of $X$ if $\forall x \in X \exists C \in \mathcal{C}$ s.t. $x \cap C \neq \emptyset$.*

**Definition 2.** *$\mathcal{C} \subset 2^S$ is a '$d_0$-cover' of a set $X$ if $\mathcal{C}$ is a cover of $X$ and if $\forall C \in \mathcal{C}$ diameter$(C) < d_0$, where diameter$(C)$ is* $\max_{c_1,c_2 \in C} d(c1, c2)$.

**Definition 3.** *A 'minimum-$d_0$-cover' is a $d_0$-cover with a minimal number of elements. We shall denote a minimum-$d_0$-cover and its size by $\mathcal{C}_{d_0}(X)$ and $c_{d_0}(X)$, respectively.*

If, for example, $X$ is a set of feature vectors in a Euclidian space, $c_{d_0}(X)$ is the minimum number of $m$-spheres with diameter $d_0$ required to cover all points in $X$.

**Definition 4.** *Given a search task $(X, l)$, let the 'max-min-target-distance', denoted $d_T$, be the largest distance of a target to its nearest non-target neighbor.*

**Theorem 1.** *Let $\mathcal{X}_{d_0,c}$ denote all the family of search tasks $(X, l)$ for which $d_T$,the max-min-target-distance, is bounded from below by some $d_0$ ($d_T \geq d_0$) and for which the minimum-$d_0$-cover size is $c$ ($c_{d_0}(X) = c$). The value $c$ quantitatively describes the difficulty of $\mathcal{X}_{d_0,c}$ in the sense that:*

1. *Any search algorithm $A$ needs to query the oracle for at least $c$ candidates in the worst case before finding a target. ($\forall A\ \exists (X, l) \in \mathcal{X}_{d_0,c}$; $cost_1(A, X, l) \geq c$)*
2. *There is an algorithm that, for all tasks in this family, needs no more than $c$ queries for finding the first target.($\exists A\ \forall (X, l) \in \mathcal{X}_{d_0,c}$ $cost_1(A, X, l) \leq c$)*

**Proof:** 1. We first provide such a 'worst case' $X$, and then choose the labels $l$ depending on the algorithm $A$. Choose $c$ points in the metric space, so that all the inner-point distances are at least $d_0$. Choose the $n$ candidates to be divided equally among these locations. Until a search algorithm finds the first target, it receives only *no* answers from the recognition oracle. Therefore, given a specific algorithm $A$ and the set $X$, the sequence of attended candidates may be simulated under the assumption that the oracle returns only *no* answers. Choose an assignment of labels $l$ that assigns $T$ only to the group of candidates located in the point whose first appearance in that sequence is last. $A$ will query the oracle at least $c$ times before finding a target.
2. We suggest the following simple algorithm, which suffices for the proof:
**FLNN-** Farthest Labeled Nearest Neighbor: Given a set of candidates $X = \{x_1, \ldots, x_n\}$, randomly choose the first candidate, query the oracle and label this candidate. Repeat iteratively, until a target is detected: for each unlabeled candidate $x_i$, compute the distance $dL_i$ to the nearest labelled neighbor. Choose the candidate $x_i$ for which $dL_i$ is maximum. Query the oracle to get its label.
Let us show that FLNN finds the first target after at most $c$ queries for all search tasks $(X, l)$ from the family $\mathcal{X}_{d_0,c}$: Take an arbitrary minimum-$d_0$-cover of $X$, $\mathcal{C}_{d_0}(X)$. Let $x_i$ be a target so that $d(x_i, x_j) \geq d_0$ for every distractor $x_j$ (such a $x_i$ exists since $d_T \geq d_0$). Let $C$ be a covering element($C \in \mathcal{C}_{d_0}(X)$) so that $x_i \in C$. Note that all candidates in $C$ are targets. Excluding $C$, there are $(c-1)$ other covering elements in $\mathcal{C}_{d_0}(X)$ with diameter $< d_0$. Since $C$ contains a candidate whose distance from all distractors $\geq d_0$, FLNN will not query two distractor-candidates in one covering element (whose distance $< d_0$), before it

queries at least one candidate in $C$. Therefore, a target will be located after at most $c$ queries. (It is possible that a target that is not in $C$ will be found earlier, and then the algorithm stops even earlier.) ∎

Note that no specific metric is considered in the above claim and proof. However, the cover size and the implied search difficulty depend on the partial description (features), which may be chosen depending on the application.

Note also that FLNN does not need to know $d_0$ and performs optimally (in the worst case) relative to the (unknown) difficulty of the task.

Note that $c_{d_T}(X)$ is the tightest suggested upper-bound on the performance of FLNN for a task $(X, l)$ for which its max-min-target-distance is $d_T$. Given a search task, naturally, we do not know who the targets are in advance and do not know $d_T$. Nevertheless, we might know that the task belongs to a family of search tasks for which $d_T$ is greater than some $d_0$. In this case we can compute $c_{d_0}(X)$, and predict an upper-bound on the queries required for FLNN.

The problem of finding the minimum cover is NP-hard. Gonzalez [4] proposes a 2-approximation algorithm for the problem of clustering a data set minimizing the maximum inner-cluster distance, and proves it is the best approximation possible if $P \neq NP$. In our experiments we used a heuristic algorithm that provided tighter upper bounds on the minimum cover size. Note also that according to the theorem, FLNN's worst cases' results may serve as a lower bound on the minimum cover size as well.

Since computing the cover is hard, we also suggest a more simple measure for search difficulty. Given a bounded metric-space containing the candidates, cover all the space with covering elements with diameter $d_0$. (For the m-dimensional bounded Euclidean metric space $[0, 1]^m$, there are $\lceil \frac{\sqrt{m}}{d_0} \rceil^m$ such elements.) The number of non-empty such covering elements is an upper-bound on the minimal cover size. See [2] for more results and a more detailed discussion.

## 4 Dynamic Search Algorithm Based on a Stochastic Model

The FLNN algorithm suffers from several drawbacks. It relates only to the nearest neighbor, which makes it non-robust. A single attended distractor close to an undetected target, reduces the priority of this target and slows the search. Moreover, it does not extend naturally to finding more than one target, and to incorporating bottom-up and top-down information, when available. The alternative algorithm suggested below addresses these problems.

### 4.1 Statistic Dependencies Modelling

Taking a stochastic approach, we model the object identities as binary random variables with possible values 0 (for non-target) or 1 (for target).

Recall that objects associated with similar identities tend to be more visually similar than objects which are of different identities. To quantify this intuition, we set the covariance between two labels to be a monotonic descending function

$\gamma$ of the feature-space-distance between them: $cov(l(x_i), l(x_j)) = \gamma(d(x_i, x_j))$, where $X = \{x_1, x_2, \ldots, x_n\}$ is a set of partial descriptions (feature vectors) associated with the set of candidates, $l(x_i)$ is the identity label of the candidate $x_i$, and $d$ is a metric distance function. In our experiments we use an exponentially descending function ($e^{-d(x_i,x_j)/d_{max}}$, where $d_{\max}$ is the greatest distance between feature-vectors), which seems to be a good approximation to the actual dependency (see Sect. 5.2).

### 4.2 Dynamic Search Framework

We propose a greedy approach to a dynamic search. At each iteration, estimate the probability of each unlabelled candidate to be a target using all the knowledge available. Choose the candidate for which the estimated probability is the highest and apply the object recognition oracle on the corresponding sub-image.

After the $m-$th iteration, $m$ candidates, $x_1, x_2, \ldots, x_m$, were already handled and $m$ labels, $l(x_1), l(x_2), \ldots, l(x_m)$ are known. We use these labels to estimate the conditional probability of the label $l(x_k)$ of each unlabelled candidate $x_k$ to be 1.

$$p_k \triangleq p(l(x_k) = 1 \mid l(x_1), \ldots, l(x_m)). \tag{1}$$

### 4.3 Minimum Mean Square Error Linear Estimation

Now, note that the random variable $l_k$ is binary and, therefore, its expected value is equal to its probability to take the value 1. Estimating the expected value, conditioned on the known data, is generally a complex problem and requires knowledge about the labels' joint distribution. We use a linear estimator minimizing the mean square error criterion, which needs only second order statistics.

Given the measured random variables $l(x_1), l(x_2), \ldots, l(x_m)$, we seek a linear estimate $\hat{l_k}$ of the unknown random variable $l(x_k)$, $\hat{l_k} = a_0 + \sum_{i=1}^{m} a_i l(x_i)$, which minimizes the minimum mean square error $e = E((l(x_k) - \hat{l_k})^2)$. Solving a set of (Yule-Walker) equations [13] provides the following estimation:

$$\hat{l_k} = E[l(x_k)] + \boldsymbol{a}^t(\boldsymbol{l} - E[\boldsymbol{l}]), \tag{2}$$

where $\boldsymbol{l} = (l(x_1), l(x_2), \ldots, l(x_m))$ and $\boldsymbol{a} = R^{-1} \cdot \boldsymbol{r}$. $R_{ij}$, $i, j = 1, \ldots, m$ and $r_i$, $i = 1, \ldots, m$ are given by $R_{ij} = \mathrm{cov}(l(x_i), l(x_j))$ and $r_i = \mathrm{cov}(l(x_k), l(x_i))$.

$E(l_k)$ is the expected value of the label $l_k$, which is the prior probability for $x_k$ to be a target. If there is no such knowledge, $E(l_k)$ can be set to be uniform, i.e., $\frac{1}{n}$ (where $n$ is the number of candidates). If there is prior knowledge on the number of targets in the scene, $E(l_k)$ should be set to $\frac{m}{n}$ (where $m$ is the expected number of targets).

The estimated label $\hat{l_k}$ is the conditional mean of a label $l(x_k)$ of an unclassified candidate $x_k$, and, therefore, may be interpreted as the probability of $l(x_k)$ to be 1

$$p_k = p(l(x_k) = T \mid l(x_1), \ldots, l(x_m)) \sim \hat{l_k}.$$

### 4.4 The Algorithm: Visual Search Using Linear Estimation – VSLE

- *Given a scene image, choose n sub-images to be candidates.*
- *Extract the set of feature vectors* $X = \{x_1, x_2, \ldots x_n\}$.
- *Calculate pairwise feature space distances and the implied covariances.*
- *Select the first candidate/s randomly (or based on some prior knowledge).*
- *In iteration* $m + 1$*:*
  - *For each candidate* $x_k$ *out of the* $n - m$ *remaining candidates, estimate* $\hat{l_k} \in [0, 1]$ *based on the known labels* $l(x_1), \ldots, l(x_m)$ *using equation 2.*
  - *Query the oracle on the candidate* $x_k$ *for which* $\hat{l_k}$ *is maximum.*
  - *If enough targets were found - abort.*

Our goal is to minimize the expected search time, and the proposed algorithm, being greedy, cannot achieve an optimal solution. It is, however, optimal with respect to all other greedy methods (based on second order statistics), as it uses all the information collected in the search to make the decision.

Note that clustered non-targets accelerate the search and even let the target pop-out when there is only a single non-target cluster. Clustered targets are found immediately after the first target is detected.

As the covariance decreases with distance, estimating the labels only from their nearest (classified) neighbors is a valid approximation which accelerates the search.

### 4.5 Combining Prior Information

Bottom-up and top-down information may be naturally integrated by specifying the prior probabilities (or the prior means) according to either the saliency or the similarity to known models. Moreover, if the top-down information is available as $k$ model images (one or more), we can simply add them as additional candidates that were examined before the actual search. Continuing the search from this point is naturally faster; see end of Sect.5.2.

## 5 Experiments

In order to test the ideas described so far, we conducted many experiments using images of different types, using different methods for candidates selection, and different features to partially describe the candidates. Below, we describe a few examples that demonstrate the relation between the algorithms' performance and the tasks' difficulty.

### 5.1 FLNN and Minimum-Cover-Size

The first set of experiments considers several search tasks and focus on their characterization using the proposed metric cover. Because calculating the minimal cover size is computationally hard, we suggest several ways to bound it

from above and from below and show that combining these methods yields a very good approximation. In this context we also test the FLNN algorithm and demonstrate its guaranteed performance. Finally we provide the intuition explaining why indeed harder search tasks are characterized by larger covers.

The first three search tasks are built around the 100 images corresponding to the 100 objects in the COIL-100 database [12] in a single pose. We think of these images as candidates extracted from some larger image. The extracted features are first, second, and third Gaussian derivatives in five scales [14] resulting in feature vectors of length 45. A Euclidian metric is used as the feature space distance. The tasks differ in the choice of the target which was cups (10 targets), toy cars (10 targets) and toy animals (7 targets) in the three search tasks.

The minimal cover size for every task is bounded as follows: First the minimal target-distractor distance, $d_T$, is calculated. We developed a greedy heuristic algorithm which prefers sparse regions and provide a possibly non-tight but always valid $d_T$- cover ; see [2] for details. For the cups search task the cover size was, for example, 24. For all tasks, this algorithm provided smaller (and tighter) covers than those obtained with the 2-approximation algorithm suggested by Gonzalez [4], which for the cups task gave a cover of size 42. Both algorithms provide upper bounds on the size of the minimal cover. See table 1 for cover sizes. Being a rigorous 2-approximation, half of the latter upper bound value (42/2=21 for the cups) is also a rigorous lower bound on the minimal cover size. Another lower bound may be found by running the FLNN algorithm itself, which, by theorem 1, needs no more than $c_{d_T}(X)$ queries to the oracle. By running the algorithm 100 times, starting from a different candidate each run and taking the largest number of queries required (18 for the cups task), we get the tightest lower bound; see table 1 where the average number of queries required by the FLNN is given as well.

Note that the search for cars was the hardest. While the car targets are very similar to each other (which should ease the search), finding the first car is hard due to the presence of distractors which are very similar to the cars ($d_T$ is small). The cups are also similar to each other, but are dissimilar to the distractors, implying an easier search. On the other hand, the different *toy animals* are dissimilar, but as one of them is very dissimilar from all candidates, the task is easier as well. Note that the minimal cover size captures the variety of reasons characterizing search difficulty in a single scalar measure.

We also experimented with images from the Berkeley hand segmented database [10] and used the segments as candidates; see Fig.1. Small segments are ignored, leaving us with 24 candidates in the *elephants* image and 30 candidates in the *parasols* image. The targets are the segments containing elephants and parasols, respectively. For those colored images we use color histograms as feature vectors. In each segment (candidate), we extract the values of $\frac{b}{r+g+b}$ and $\frac{r}{r+g+b}$ from each pixel, where $r$, $g$, and $b$ are values from the RGB representation. Each of these two dimensions is divided into 8 bins, resulting a feature vector of length 64. Again, we use Euclidean metric for distance measure. (Using other histogram comparison methods, such as the ones suggested in [16] the results

**Fig. 1.** The *elephants* and *parasols* images taken from the Berkeley hand segmented database and the segmentations we used in our experiments. (colored images)

**Table 1.** Experiment results for FLNN and cover size. The real value of minimal cover size is bounded from below by 'FLNN worst' and the half of '2-Approx. cover size', and bounded from above by 'Heuristic cover size' and '2-Approx. cover size'. The rightmost column shows that VSLE improves the results of FLNN for finding the first target.

| Search task | # of cand. | # of targets | FLNN worst | FLNN mean | Heuristic cover size | 2-Approx. cover size | Real cover size | VSLE worst |
|---|---|---|---|---|---|---|---|---|
| cups | 100 | 10 | 18 | 8.97 | 24 | 42 | 21-24 | 15 |
| cars | 100 | 10 | 73 | 33.02 | 79 | 88 | 73-79 | 39 |
| toy animals | 100 | 7 | 22 | 9.06 | 25 | 42 | 22-25 | 13 |
| elephants | 24 | 4 | 9 | 5.67 | 9 | 11 | 9 | 8 |
| parasols | 30 | 6 | 6 | 3.17 | 8 | 13 | 7-8 | 4 |

were similar.) See the results in Table 1. Although the mean results are usually not better than the mean results of a random search, the worst results are much better.

### 5.2 VSLE and Covariance Characteristics

The VSLE algorithm described in Sect.4 was implemented and applied to the same five visual search tasks described in Sect.5.1. See Fig.2 for part of the results. Unlike FLNN which deals only with finding the first target, VSLE continues and aims also to find the other targets. Moreover, in almost all the experiments we performed, VSLE was faster in finding the first target (both in the worst and the mean results). See the rightmost column in table 1.

VSLE relies on the covariance between candidates' labels. We use a covariance function that depends only on feature-space-distance, and argue that for many search tasks this function is monotonic descending in this distance. To check this assumption we estimate the covariance of labels vs. feature-space-distance of search tasks and confirmed for its validity; see Fig.2 and [2].

We experimented with a preliminary version of integrated segmentation and search. An input image (see Fig.3) was segmented using k means clustering in the RGB color space (using 6 clusters). All (146) connected components larger than 100 pixels served as candidates. The VSLE algorithm searched for the (7)

faces in the image, using a feature vector of length 4: each segment is represented by the mean values of red green and blue and the segment size.

No prior information on size, shape color or location was used. Note that this search task is hard due to the presence of similarly colored objects in the background, and due to the presence of hands which share the same color but are not classified as targets. Note that in most runs six of the seven faces are detected after about one-sixth of the segments are examined. We deliberately chose a very crude segmentation, demonstrating that very good segmentation is not required for the proposed search mechanism.

Using the method suggested in Sect.4.5, we incorporate top-down information and demonstrate it on the *toy cars* case: 3 toy cars which do not belong to the COIL-100 database are used as model targets. The search time was significantly reduced as expected; see Fig.4.

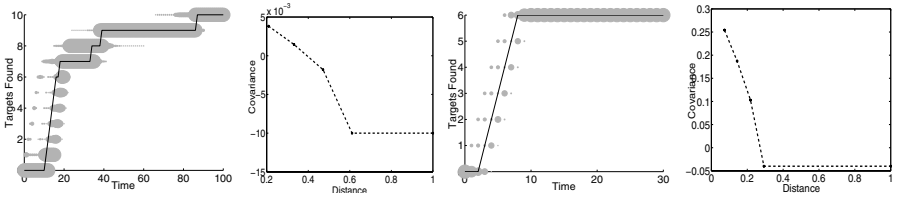
## 6 Discussion

In this paper we considered the usage of inner-scene similarity for visual search, and provided both measures for the difficulty of the search, and algorithms for implementing it. We took a quantitative approach, allowing us not only to optimize the search but also to quantitatively predict its performance.

Interestingly, while we did not aim at modelling the HVS attention system, it turns out that it shares many of its properties, and in particular, is similar to Duncan and Humphreys's model [3]. As such, our work can be considered as a quantification of their observations. Not surprisingly, our results also show that there is a continuity between the two poles of 'pop-out' and 'sequential' searches.

While many search tasks rely only on top down or bottom up knowledge, inner scene similarities always help and may become the dominant source of knowledge when less is known about the target. Consider, for example the *parasols* search task (Sect. 5). First, note that the targets take a significant image fraction, and cannot be salient. Then, the parasols are similar and different from the non-targets in their color, but if this color is unknown, they cannot be searched using top-down information. More generally, considering a scene containing several objects of the same category, we argue that their sub-images are more similar than images of such objects taken in different times and places. This happens because the imaging conditions are more uniform and because the variability of objects is smaller in one place. (e.g, two randomly chosen trees are more likely to be of the same type if they are taken from the same area.)

We are now working on building an overall automatic system that will combine the suggested algorithms (extended to use bottom-up and top-down information) with grouping and object recognition methods. We also intend to continue analyzing search performance. We would like to be able to predict search time for the VSLE algorithm, for instance, in a manner similar to that we have achieved for FLNN. While the measure of minimal cover size as a lower bound for the worst cases holds, we aim to suggest a tighter bound for cases that are statistically more common.

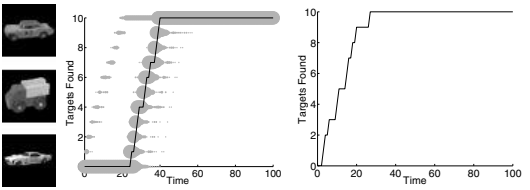



*Cups* VSLE (a) *Cups* covariance (b) *Parasols* VSLE (c) *Parasols* covariance (d)

**Fig. 2.** VSLE and covariance vs. distance results. (a) VSLE results for the *cups* search task.The solid lines describe one typical run. Other runs, starting each time from a different candidate, are described by the size of the gray spots as a distribution in the (time, number of targets found) space. It is easy to find the first cup since most cups are different from non-targets. Most cups resemble and follow pretty fast, but there are three cups (two without a handle and one with a special pattern) that are different from the rest of the cups, and are found rather late. (b) Estimate of labels covariance vs. feature-space-distance for the *cups* search task. (c) VSLE results for the *parasols* search task. All the parasols are detected very fast, since their color is similar and differs from that of all other candidates. (d) Estimate of labels covariance vs. feature-space-distance for the *parasols* search task.

(a) (b) (c)

**Fig. 3.** VSLE applied on an automatic-color-segmented image to detect faces. (a) The input image (colored image) (b) Results of an automatic crude color-based segmentation (c) VSLE results (see caption of figure 2 for what is shown in this graph).




(a) (b) (c)

**Fig. 4.** VSLE using top-down information for the *toy cars* search task. (a) The three model images. (b) VSLE results without using the models, (c)results of extended VSLE using the model images.

## References


1. T. Avraham and M. Lindenbaum. A Probabilistic Estimation Approach for Dynamic Visual Search. *Proceedings of International Workshop on Attention and Performance in Computer Vision (WAPCV)*, 1–8, 2003.
2. T. Avraham and M. Lindenbaum. CIS Report #CIS-2003-02, 2003. Technion - Israel Institute of Technology, Haifa 32000, Israel.
3. J. Duncan and G.W. Humphreys. Visual search and stimulus similarity. *Psychological Review*, 96:433–458, 1989.
4. T.F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(2-3):293–306, June 1985.
5. G.W. Humphreys and H.J. Muller. Search via recursive rejection (serr): A connectionist model of visual search. *Cognitive Psychology*, 25:43–110, 1993.
6. L. Itti. Models of bottom-up and top-down visual attention. *Thesis*, January 2000.
7. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259, November 1998.
8. C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural vircuity. *Human Neurobiology*, 4:219–227, 1985.
9. A.N. Kolmogorov and V.M. Tikhomirov. *epsilon*-entropy and *epsilon*-capacity of sets in functional spaces. *AMS Translations. Series 2*, 17:277–364, 1961.
10. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th ICCV*, volume 2, pages 416–423, July 2001.
11. U. Neisser. *Cognitive Psychology*. Appleton-Century-Crofts, New York, 1967.
12. S. Nene, S. Nayar, and H. Murase. Columbia object image library (coil-100). *Technical Report CUCS-006-96, Department of Computer Science, Columbia University*, February 1996.
13. A. Papoulis and S.U. Pillai. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York, NY, USA, fourth edition, 2002.
14. R.P.N. Rao and D.H. Ballard. An active vision architecture based on iconic representations. *Artificial Intelligence*, 78(1–2):461–505, 1995.
15. R.D. Rimey and C.M. Brown. Control of selective perception using bayes nets and decision theory. *International Journal of Computer Vision*, 12:173–207, 1994.
16. M.J. Swain and D.H. Ballard. Color indexing. *IJCV*, 7:11–32, 1991.
17. H. Tagare, K. Toyama, and J.G. Wang. A maximum-likelihood strategy for directing attention during visual search. *IEEE PAMI*, 23(5):490–500, 2001.
18. A. Torralba and P. Sinha. Statistical context priming for object detection. In *Proceedings of the 8th ICCV*, pages 763–770, 2001.
19. A. Treisman and G.Gelade. A feature integration theory of attention. *Cognitive Psychology*, 12:97–136, 1980.
20. J.K. Tsotsos. On the relative complexity of active versus passive visual search. *IJCV*, 7(2):127–141, 1992.
21. J.K. Tsotsos, S.M. Culhane, W.Y.K. Wai, Y. Lai, N. Davis, and F.J. Nuflo. Modeling visual attention via selective tuning. *Artificial intelligence*, 78(1-2):507–545, 1995.
22. L.E. Wixson and D.H. Ballard. Using intermediate objects to improve the efficiency of visual-search. *IJCV*, 12(2-3):209–230, April 1994.
23. J.M. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1(2):202–238, 1994.
24. A.L. Yarbus. *Eye Movements and Vision*. Plenum Press, New York, 1967.

# Weak Hypotheses and Boosting for Generic Object Detection and Recognition

A. Opelt[1,2], M. Fussenegger[1,2], A. Pinz[2], and P. Auer[1]

[1] Institute of Computer Science,
8700 Leoben, Austria
{auer,andreas.opelt}@unileoben.ac.at
[2] Institute of Electrical Measurement and Measurement Signal Processing,
8010 Graz, Austria
{fussenegger,opelt,pinz}@emt.tugraz.at

**Abstract.** In this paper we describe the first stage of a new learning system for object detection and recognition. For our system we propose Boosting [5] as the underlying learning technique. This allows the use of very diverse sets of visual features in the learning process within a common framework: Boosting — together with a weak hypotheses finder — may choose very inhomogeneous features as most relevant for combination into a final hypothesis. As another advantage the weak hypotheses finder may search the weak hypotheses space without explicit calculation of all available hypotheses, reducing computation time. This contrasts the related work of Agarwal and Roth [1] where Winnow was used as learning algorithm and all weak hypotheses were calculated explicitly. In our first empirical evaluation we use four types of local descriptors: two basic ones consisting of a set of grayvalues and intensity moments and two high level descriptors: moment invariants [8] and SIFTs [12]. The descriptors are calculated from local patches detected by an interest point operator. The weak hypotheses finder selects one of the local patches and one type of local descriptor and efficiently searches for the most discriminative similarity threshold. This differs from other work on Boosting for object recognition where simple rectangular hypotheses [22] or complex classifiers [20] have been used. In relatively simple images, where the objects are prominent, our approach yields results comparable to the state-of-the-art [3]. But we also obtain very good results on more complex images, where the objects are located in arbitrary positions, poses, and scales in the images. These results indicate that our flexible approach, which also allows the inclusion of features from segmented regions and even spatial relationships, leads us a significant step towards generic object recognition.

## 1 Introduction

We believe that a learning component is a necessary part of any generic object recognition system. In this paper we investigate a principle approach for learning objects in still images which allows the use of flexible and extendible

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 71–84, 2004.
© Springer-Verlag Berlin Heidelberg 2004

sets of features for describing objects and object categories. Objects should be recognized even if they occur at abitrary scale, shown from different perspective views on highly textured backgrounds. Our main learning technique relies on Boosting [5]. Boosting is a technique for combining several weak classifiers into a final strong classifier. The weak classifiers are calculated on different weightings of the training examples to emphasize different portions of the training set. Since any classification function can potentially serve as a weak classifier we can use classifiers based on arbitrary and inhomogeneous sets of image features. A further advantage of Boosting is that weak classifiers may be calculated when needed instead of calculating unnecessary hypotheses a priori.

In our learning setting, the learning algorithm needs to learn an object category. It is provided with a set of labeled training images, where a positive label indicates that a relevant object appears in the image. The objects are not segmented and pose and location are unknown. As output, the learning algorithm delivers a final classifier which predicts if a relevant object is present in a new image. Having such a classifier, the localization of the object in the image is straightforward. The image analysis transforms images to greyvalues and extracts normalised regions around interest (salient) points to obtain reduced representations of images. As an appropriate representation for the learning procedure we calculate local descriptors of these patches. The result of the training procedure is saved as the final hypothesis which is later used for testing (see figure 1).

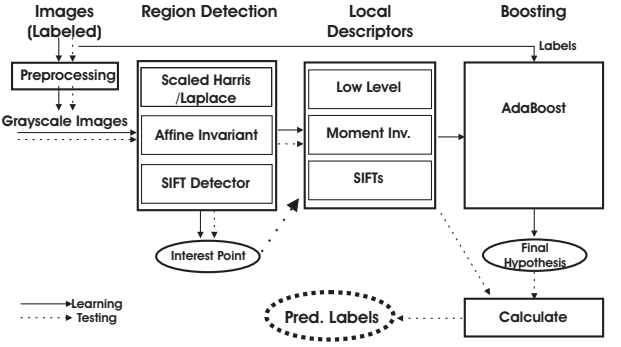



**Fig. 1.** Overview showing the framework for our approach for generic object recognition. The solid arrows show the training cycle, the dotted ones the testing procedure.

We describe our general learning approach in detail in section 2. In section 3, we discuss the image analysis steps, including illumination and size normalisation, interest point detection, and the extraction of the local descriptors. An explicit explanation of how we calculate the weak hypotheses used by the Boosting algorithm, is given in section 4. Section 5 contains a description of the setup we used for our experiments. The results are presented and compared with other

approaches for object recognition. We regard the present system as a first step and further work is outlined in section 6.

### 1.1 Related Work

Clearly there is an extensive body of literature on object recognition (e.g. [3], [2], [22], [24], [6], [14]). In general, these approaches use image databases which show the object of interest at prominent scales and with only little variation in pose. We discuss only some of the most relevant and most recent results related to our approach.

Boosting was successfully used by Viola and Jones [22] as the ingredient for a fast face detector. The weak hypotheses were the thresholded average brightness of collections of up to four rectangular regions. In our approach we experiment with much larger sets of features to be able to perform recognition of a wider class of objects.

Schneiderman and Kanade [20] used Boosting to improve an already complex classifier. In contrast, we are using Boosting to combine rather simple classifiers by selecting the most discriminative features.

Agarwal and Roth [1] used Winnow as the underlying learning algorithm for the recognition of cars from side views. For this purpose images were represented as binary feature vectors. The bits of such a feature vector can be seen as the outcomes of weak classifiers, one weak classifier for each position in the binary vector. Thus for learning it is required that the outcomes of all weak classifiers are calculated a priori. In contrast, Boosting only needs to find the few weak classifiers which actually appear in the final classifier. This substantially speeds up learning, if the space of weak classifiers carries a structure which allows the efficient search for discriminative weak classifiers. A simple example is a weak classifier which compares a real valued feature against a threshold. For Winnow, one weak classifier needs to be calculated for each possible threshold a priori[1], whereas for Boosting the optimal threshold can be determined efficiently when needed.

A different approach to object class recognition was presented by Fergus, Perona, and Zisserman [3]. They used a generative probabilistic model for objects built as constellations of parts. Using an EM-type learning algorithm they achieved very good recognition performance. In our work we have chosen a model-free approach for flexibility. If at all, the sets of weak classifiers we use can be seen as model classes, but with much less structure than in [3]. Furthermore, we propose Boosting as a very different learning algorithm from EM.

Dorko and Schmid [2] introduced an approach for constructing and selecting scale-invariant object parts. These parts are subsequently used to learn a classifier. They show a robust detection under scale changes and variations in viewing conditions, but in contrast to our approach, the objects of interest are manually pre-segmented. This dramatically reduces the complexity of distinguishing between relevant patches on the objects and background clutter.

[1] More efficient techniques for Winnow like using virtual threshold gates [13] do not improve the situation much.

## 2 Our Learning Model for Object Recognition

In our setup, a learning algorithm has to recognize objects from a certain category in still images. For this purpose, the learning algorithm delivers a classifier that predicts whether a given image contains an object from this category or not. As training data, labeled images $(I_1, \ell_1), \ldots, (I_m, \ell_m)$ are provided for the learning algorithm where $\ell_k = +1$ if $I_k$ contains a relevant object and $\ell_k = -1$ if $I_k$ contains no relevant object. Now the learning algorithm delivers a function $H : I \mapsto \hat{\ell}$ which predicts the label of image $I$. To calculate this classification function $H$ we use the classical AdaBoost algorithm [5]. AdaBoost puts weights $w_k$ on the training images and requires the construction of a weak hypothesis $h$ which has some discriminative power relative to these weights, i.e.

$$\sum_{k:h(I_k)=\ell_k} w_k > \sum_{k:h(I_k)\neq\ell_k} w_k \ , \tag{1}$$

such that more images are correctly classified than misclassified, relative to the weights $w_k$. (Such a hypothesis is called weak since it needs to satisfy only a very weak requirement.) The process of putting weights and constructing a weak hypothesis is iterated for several rounds $t = 1, \ldots, T$, and the weak hypotheses $h_t$ of each round are combined into the final hypothesis $H$.

In each round $t$ the weight $w_k$ is decreased if the prediction for $I_k$ was correct ($h_t(I_k) = \ell_k$), and increased if the prediction was incorrect. Different to the standard AdaBoost algorithm we vary the factor $\beta_t$ to trade off precision and recall. We set

$$\beta_t = \begin{cases} \sqrt{\frac{1-\varepsilon}{\varepsilon}} * \eta & \text{if } \ell_k = +1 \text{ and } \ell_k \neq h_t(I_k). \\ \sqrt{\frac{1-\varepsilon}{\varepsilon}} & \text{else} \end{cases}$$

with $\varepsilon$ being the error of the weak hypothesis in this round and $\eta$ as an additional weight factor to control the update of wrongly classified positive examples.

Here two general comments are in place. First, it is intuitively quite clear that weak hypotheses with high discriminative power — with a large difference of the sums in (1) — are preferable, and indeed this is shown in the convergence proof of AdaBoost [5]. Second, the adaptation of the weights $w_k$ in each round performs some sort of adaptive decorrelation of the weak hypotheses: if an image was correctly classified in round $t$, then its weight is decreased and less emphasis is put on this image in the next round, yielding quite different hypotheses $h_t$ and $h_{t+1}$.[2] Thus it can be expected that the first few weak hypotheses characterize the object category under consideration quite well. This is particularly interesting when a sparse representation of the object category is needed.

Obviously AdaBoost is a very general learning technique for obtaining classification functions. To adapt it for a specific application, suitable weak hypotheses

[2] In fact AdaBoost sets the weights in such a way that $h_t$ is *not* discriminative in respect to the *new* weights. Thus $h_t$ is in some sense oblivious to the predictions of $h_{t+1}$.

have to be constructed. For the purpose of object recognition we need to extract suitable features from images and use these features to construct the weak hypotheses. Since AdaBoost is a general learning technique we are free to choose any type of features we like, as long as we are able to provide an effective weak hypotheses finder which returns discriminative weak hypotheses based on this set of features. The chosen features should be able to represent the content of images, at least in respect to the object category under consideration. Since we may choose several types of features, we represent an image $I$ by a set of pairs $\mathcal{R}(I) = \{(\tau, v)\}$ where $\tau$ denotes the type of a feature and $v$ denotes a value of this feature, typically a vector of reals. Then for AdaBoost a weak hypothesis is constructed from the representations $\mathcal{R}(I_k)$, labels $\ell_k$, and weights $w_k$ of the training images.

In the next section we describe the types of features we are currently using, although many other features could be used, too. In Section 4 we describe the effective construction of the weak hypotheses.

## 3 Image Analysis and Feature Construction

We extract features from raw images, ignoring the labels used for learning. To lower the number of the points in an image we have to attend to, we use an interest point detector to get salient points. We evaluate three different detectors, a scale invariant interest point detector, an affine invariant interest point detector, and the SIFT interest point detector ([15], [16], [12], see section 3.1). Using these salient points we can reduce the content of an image to a number of points (and their surroundings) while being robust against irrelevant variations in illumination and scale. Since the most salient points[3] may not belong to the relevant objects, we have to take a rather large number of points into account, which implies choosing a low threshold in the interest point detectors. The number of SIFTs is reduced by a vector quantization using k-means (similarly to Fergus et al. [3]). The pixels enclosing an interest point are refered to as a patch. Due to different illumination conditions we normalise each patch before the local descriptors are calculated. Representing patches through a local descriptor can be done in different ways. We use subsampled grayvalues, intensity moments, Moment Invariants and SIFTs here.

### 3.1 Interest Point Detection

There is a variety of work on interest point detection at fixed (e.g. [9,21,25,10]), and at varying scales (e.g. [11,15,16]). Based on the evaluation of interest point detectors by Schmid et al. [19], we decided to use the scale invariant Harris-Laplace detector [15] and the affine invariant interest point detector [16], both by Mikolajczyk and Schmid. In addition we use the interest point detector used by Lowe [12] because it is strongly interrelated with SIFTs as local descriptors.

[3] E.g. by measuring the entropy of the histogram in the surrounding [3] or doing a Principal Component Analysis.

The scale invariant detector finds interest points by calculating a scaled version of the second moment matrix $M$ and localizing points where the Harris Measure $H = det(M) - \alpha trace^2(M)$ is above a certain threshold $th$. The characteristic scale for each of these points is found in scale-space by calculating the Laplacians $L(\mathbf{x}, \sigma) = |\sigma^2(L_{xx}(\mathbf{x}, \sigma) + L_{yy}(\mathbf{x}, \sigma))|$ for each desired scale $\sigma$ and taking the one at which $L$ has a maximum in an 8-neighbourhood of the point.

The affine invariant detector is also based on the second moment matrix computed at a point which can be used to normalise a region in an affine invariant way. The characteristic scale is again obtained by selecting the scale at which the Laplacian has a maximum. An iterative algorithm is then used which converges to affine invariant points by modifying the location, scale and neighbourhood of each point.

Lowe introduced an interest point detector invariant to translation, scaling and rotation and minimally affected by small distortions and noise [12]. He also uses the scale-space but built with a difference of Gaussian (DoG). Additionally, a scale pyramid achieved by bilinear interpolation is employed. Calculating the image gradient magnitude and the orientation at each point of the scale pyramid, salient points with characteristic scales and orientations are achieved.

### 3.2 Region Normalisation

To normalise the patches we have to consider illumination, scale and affine transformations. For the size normalisation we have decided to use quadratic patches with a side of $l$ pixels. The value of $l$ is a variable we vary in our experiments. We extract a window of size $w = 6 * \sigma_I$ where $\sigma_I$ is the characteristic scale of the interest point delivered by the interest point detector. Scale normalisation is done by smoothing and subsampling in cases of $l < w$ and by linear interpolation otherwise. In order to obtain affine invariant patches the values of the transformation matrix resulting from the affine invariant interest point detector are used to normalise the window to the shape of a square, before the size normalisation.

For illumination normalisation we use Homomorphic Filtering (see e.g. [7], chapter 4.5). The Homomorphic Filter is based on an image formation model where the image intensity $I(x, y) = i(x, y)r(x, y)$ is modeled as the product of illumination $i(x, y)$ and reflectance $r(x, y)$. Elimination of the illumination part leads to a normalisation. This is achieved by applying a Fast Fourier Transform to the logarithm image $ln(I)$. Now the reflectance component can be separated by a high pass filter. After a back transformation and an exponentiation we get the desired normalised patch.

### 3.3 Feature Extraction

To represent each patch we have to choose some local descriptors. Local descriptors have been researched quite well (e.g. [4], [12], [18], [8]). We selected four local descriptors for our patches. Our first descriptor is simply a vector of all pixels in a patch subsampled by two. The dimension of this vector is $\frac{l}{4}^2$ which is rather high and increases computational complexity. As a second descriptor we use intensity moments $M^a_{I_{pq}} = \int\int_\omega i(x, y)^a x^p y^q \, dx \, dy$ with $a$ as the degree and

$p + q$ as the order, up to degree 2 and order 2. Without using the moments of degree 0 we get a feature vector with a dimension of 10. This reduces the computational costs dramatically. With respect to the performance evaluation of local descriptors done by Mikolajczyk and Schmid [17] we took SIFTs (see [12]) as a third and Moment Invariants (see [8]) as a fourth choice. In this evaluation the SIFTs outmatched the others in nearly all tests and the Moment Invariants were in the middle ground for all aspects considered.

According to [8] we selected first and second order Moment Invariants. We chose the first order affine Invariant and four first order affine and photometric Invariants. Additionally we took all five second order Invariants described in [8]. Since the Invariants require two contours, the whole square patch is taken as one contour and rectangles corresponding to one half of the patch are used as a second contour. All four possibilities of the second contour are calculated and used to obtain the Invariants. The dimenson of the Moment Invariants description vector is 10.

As shown in [12] the description of the patches with SIFTs is done by multiple representations in various orientation planes. These orientation planes are blurred and resampled to allow larger shifts in positions of the gradients. A local descriptor with a dimension of 128 is obtained here for a circular region around the point with a radius of 8 pixels, 8 orientation planes and sampling over a 4x4 and a 2x2 grid of locations.

## 4 Calculation of Weak Hypotheses

Using the features constructed in the previous section, an image is represented by a list of features $(\tau_f, v_f)$, $f = 1, \ldots, F$, where $\tau_f$ denotes the type of a feature, $v_f$ denotes its value as real vector, and $F$ is the number of extracted features in an image. The weak hypotheses for AdaBoost are calculated from these features. For object recognition we have chosen weak hypotheses which indicate if certain feature values appear in images. For this a weak hypothesis $h$ has to select a feature type $\tau$, its value $v$, and a similarity threshold $\theta$. The threshold $\theta$ decides if an image contains a feature value $v_f$ that is sufficiently similar to $v$. The similarity between $v_f$ and $v$ is calculated by the Mahalanobis distance for Moment Invariants and by the Euclidean distance for SIFTs. The weak hypotheses finder searches for the optimal weak hypothesis — given labeled representations of the training images $(\mathcal{R}(I_1), \ell_1), \ldots, (\mathcal{R}(I_m), \ell_m)$ and their weights $w_1, \ldots, w_m$ calculated by AdaBoost — among all possible feature values and corresponding thresholds.

The main computational burden is the calculation of the distances between $v_f$ and $v$, since they both range over all feature values that appear in the training images.[4] Given these distances which can be calculated prior to Boosting, the remaining calculations are relatively inexpensive. Details for the weak hypotheses finder are given in Figure 2. After sorting the optimal threshold for feature $(\tau_{k,f}, v_{k,f})$ can now be calculated in time $O(m)$ by scanning through the weights

[4] We discuss possible improvements in Section 6.

**Input:** Labeled representations $(\mathcal{R}(I_k), \ell_k)$,
$k = 1, \ldots, m$, $\mathcal{R}(I_k) = \{(\tau_{k,f}, v_{k,f}) : f = 1, \ldots, F_k\}$.
**Distance functions:** Let $d_\tau(\cdot, \cdot)$ be the distance in respect to the feature values of type $\tau$ in the training images.
**Minimal distance matrix:** For all features $(\tau_{k,f}, v_{k,f})$ and all images $I_j$ calculate the minimal distance between $v_{k,f}$ and features in $I_j$,

$$d_{k,f,j} = \min_{1 \le g \le F_j : \tau_{j,g} = \tau_{k,f}} d_{\tau_{k,f}}(v_{k,f}, v_{j,g}) \ .$$

**Sorting:** For each $k, f$ let $\pi_{k,f}(1), \ldots, \pi_{k,f}(m)$ be a permutation such that

$$d_{k,f,\pi_{k,f}(1)} \le \cdots \le d_{k,f,\pi_{k,f}(m)} \ .$$

**Select best weak hypothesis (Scanline):** For all features $(\tau_{k,f}, v_{k,f})$ calculate over all images $I_j$

$$\max_s \sum_{i=1}^{s} w_{\pi_{k,f}(i)} * \ell_{\pi_{k,f}(i)} \ .$$

and select the feature $(\tau_{i,f}, v_{i,f})$ where the maximum is achieved. **Select threshold $\theta$:** With the position $s$ where the scanline reached a maxium sum the threshold $\theta$ is set to

$$\theta = \frac{d_{k,f,\pi_{k,f}(s)} - d_{k,f,\pi_{k,f}(s+1)}}{2} \ .$$

**Fig. 2.** Explanation of the weak hypotheses finder.

$w_1, \ldots, w_m$ in the order of the distances $d_{k,f,j}$. Searching over all features, the calculation of the optimal weak hypothesis takes $O(Fm)$ time.

To give an example of the absolute computation times we used a dataset of 150 positive and 150 negative images. Each image has an average number of approximately 400 patches. Using SIFTs one iteration after preprocessing requires about one minute computation time on a P4, 2.4GHz PC.

## 5 Experimental Setup and Results

We carried out our experiments as follows: the whole approach was first tested on the database used by Fergus et al. [3]. After demonstrating a comparable performance, the approach was tested on a new, more difficult database[5], see figure 5. These images contain the objects at arbitrary scales and poses. The images also contain highly textured background. Testing on these images shows that our approach still performs well. We have used two categories of objects, persons (P) and bikes (B), and images containing none of these objects (N). Our database contains 450 images of category P, 350 of B and 250 of category N. The recognition was based on deciding presence or absence of a relevant object.

[5] Available at *http://www.emt.tugraz.at/~pinz/data/*

Preparing our data set we randomly chose a number of images, half belonging to the object category we want to learn and half not. From each of these two piles we take one third of the images as a set of images for testing the achieved model. The performance was measured with the receiver-operating characteristic (ROC) corresponding error rate. We tested the images containing the object (e.g. category B) against non-object images from the database (e.g. categories P and N). Our training set contains 100 positive and 100 negative images. The tests are carried out on 100 new images, half belonging to the learned class and half not. Each experiment was done using just one type of local descriptor.

Figure 3(a) shows the recall-precision curve (RPC) of our approach (obtained by varying $\eta$), the approach of Fergus et al. [3] and the one of Agarwal and Roth [1], trained on the dataset used by Fergus et al. [3][6]. Our approach performs better than the one of Agarwal and Roth but slightly worse than the approach of Fergus et al.

Table 1 shows the results of our approach (using the affine invariant interest point detection and Moment Invariants) compared with the ones of Fergus et al. and other methods [23], [24], [1]. While they use a kind of scale and viewing direction normalisation (see [3]), we work on the original images. Our results are almost as good as the results of Fergus et al. for the motorbikes dataset. For the other datasets our error rate is somewhat higher than the one of Fergus et al., but mostly lower than the error rate of the other methods.

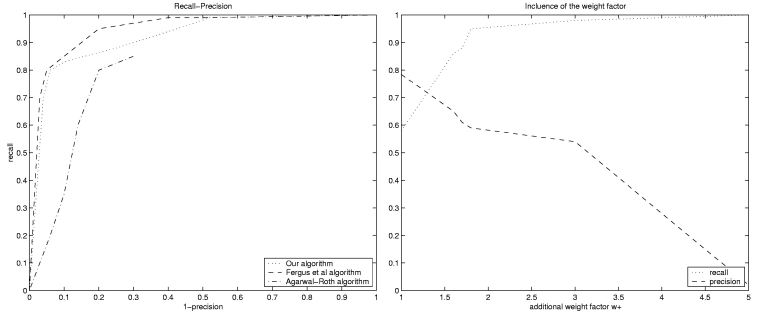
**Table 1.** The table gives the ROC equal error rates on a number of datasets from the database used by Fergus et al. [3]. Our results (using the affine invariant interest point detection and Moment Invariants) are compared with the results of the approach of Fergus et al. and other methods [23], [24], [1]. The error rates of our algorithm are between the other approaches and the ones of Fergus et al. in all cases except for the faces where the algorithm of Weber et al. [24] is also slightly better.

| Dataset | Ours | Fergus et al. [3] | Others | Ref. |
|---|---|---|---|---|
| Motorbikes | 92.2 | 92.5 | 84 | [23] |
| Airplanes | 88.9 | 90.2 | 68 | [23] |
| Faces | 93.5 | 96.4 | 94 | [24] |
| Cars(Side) | 83.0 | 88.5 | 79 | [1] |

This comparison shows that our approach performs well on the Fergus et al. database. We proceed with experiments on our own dataset and show some effects of parameter tuning[7]. Figure 3(b) shows the influence of the additional weighting of right positive examples in the Boosting algorithm ($\eta$). We can see that with a factor $\eta$ smaller than 1.8, the recall increases faster than the precision

[6] Available at $http://www.robots.ox.ac.uk/\sim vgg/data/$

[7] Parametes not given in these tests are set to $\eta = 1.8$, $T = 50$, $l = 16px$, $th = 30000$, smallest scale is skipped. Depending on textured/homogenous background, the number of interest points detected in an image varies between 50 and 1000.




(a) (b)

**Fig. 3.** The curves in (a) and (b) are obtained by varying the factor $\eta$. In (a) the diagram shows the recall-precision curve for [3], [1] and our approach on the cars (side) dataset. Our approach is superior to the one of Agarwal and Roth but slightly worse than the one of Fergus et al. The diagram (b) shows the influence of an additional factor $\eta$ for the weights of correctly positive classified examples. The recall increases faster than the precision drops until a factor of 1.8.

drops. Then both curves have nearly the same (but inverse) gradient up to a factor of 3. For $\eta > 3$ the precision decreases rapidly with no relevant gain of recall.

Table 2 presents the performance of the Moment Invarants as local descriptor, compared with our low level descriptors (using the affine invariant interest point detector). Moment Invariants delivered the best results but the other low level descriptors did not perform badly, either. This behaviour might be explained by the fact that the extracted regions are already normalised against the same set of transformations as the Moment Invariants.

**Table 2.** The table shows the results we reached with the three different kinds of local descriptors. We used an additional weight factor $\eta = 1.7$ here. Moment Invariants delivered the best results.

| Local Descriptor | recall | precision |
|---|---|---|
| Moment Invariants | 0.88 | 0.61 |
| Intensity Moments | 0.70 | 0.57 |
| Subsampled Grayvalues | 0.82 | 0.62 |

In table 3 the results of our approach using the scale invariant interest point detector compared with the use of the affine invariant interest point detector are shown. We also vary the additional weight for right positive classified examples $\eta$. The affine invariant interest point detector achieves better results for the recall but precision is higher when we use the scale invariant version of the interest

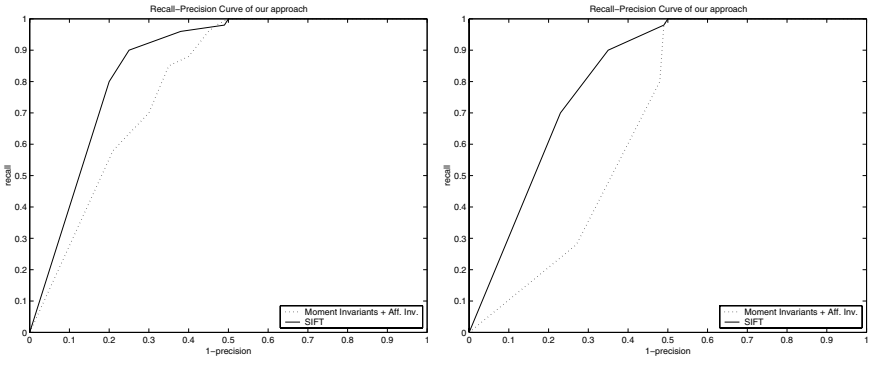



**Fig. 4.** In (a) the recall-precision curve of our approach with Moment Invariants and the affine invariant interest point detection, and the recall-precision curve of our approach using SIFTs for category bike are shown. (b) shows the recall-precision curves with the same methods for the category person.

point detector. This is to be expected since the affine invariant detector allows for more variation in the image, which implies higher recall but less precision.

**Table 3.** The table shows the results of our approach using the scale invariant interest point detector compared with using the affine invariant interest point detector varying the additional weight for right positive classified examples $\eta$.

| $\eta$ | recall (scale inv.) | precision (scale inv.) | recall (affine inv.) | precision (affine inv.) |
|---|---|---|---|---|
| 1.7 | 0.78 | 0.70 | 0.88 | 0.61 |
| 1.9 | 0.79 | 0.64 | 0.92 | 0.59 |
| 2.1 | 0.82 | 0.62 | 0.94 | 0.57 |

We skipped the smallest scale in our experiments because experiments show that this reduction of number of points does not have relevant influence to the error rates. Again, using the parameters that performed best, figure 4(a) shows an example of a recall-precision curve (RPC) of our approach trained on the bike dataset from our image database with Moment Invariants and the affine invariant interest point detection compared with our approach using SIFTs. Using the same methods we obtain the recall-precision curves (RPC) shown in figure 4(b) for the category person.

For directly comparing the results reached using the Moment Invariants with the affine invariant interest point detector or using SIFTs, the ROC equal error rates on various datasets are shown in table 4. As seen here the SIFTs perform better on our database. Tested on a category of the database from Fergus et al. one can see that the Moment Invariants perform better in that case.

**Table 4.** This table shows a comparison of the ROC equal error rates reached with the two high level features. On our database the SIFTs perform better, but on the database of Fergus et al. the Moment Invariants reach the better error rate.

| Dataset | Moment Invariants | SIFTs |
|---|---|---|
| Airplanes | 88.9 | 80.5 |
| Bikes | 76.5 | 86.5 |
| Persons | 68.7 | 80.8 |

## 6 Discussion and Outlook

In conclusion, we have presented a novel approach for the detection and recognition of object categories in still images. Our system uses several steps of image analysis and feature extraction, which have been previously described, but succeeds on rather complex images with a lot of background structure. Objects are shown in substantially different poses and scales, and in many of the images the objects (bikes or persons) cover only a small portion of the whole image. The main contribution of the paper, however, lies in the new concept of learning. We use Boosting as the underlying learning technique and combine it with a weak hypothesis finder. In addition to several other advantages of this approach, which have already been mentioned, we want to emphasize that this approach allows the formation of very diverse visual features into a final hypothesis. We think that this capability is the main reason for the good experimental results on our complex database. Furthermore, experimental comparison on the database used by Fergus et al. [3] shows that our approach performs similarly well to state-of-the-art object categorization on simpler images.

We are currently investigating extensions of our approach in several directions. Maybe the most obvious is the addition of more features to our image analysis. This includes not only other local descriptors like differential invariants [12], but also regional features[8] and geometric features[9]. To reduce the complexity of our approach we are considering a reduction of the number of features by clustering methods.

As the next step we will use spatial relations between features to improve the accuracy of our object detector. To handle the complexity of many possible relations between features, we will use the features constructed in our current approach (with parameters set for high recall) as starting points. Boosting will again be the underlying method for learning object representations as spatial combinations of features. This will allow the construction of weak hypotheses for discriminative spatial relations.

**Acknowledgements.** This work was supported by the European project LAVA (IST-2001-34405) and by the Austrian Science Foundation (FWF, project S9103-

[8] Regional features describe regions found by appearance based clustering.

[9] A geometric feature describes the appearance of geometric shapes, e.g. ellipses, in images.

**Fig. 5.** Examples from our image data base. The first column shows three images from the object class bike, the second column contains objects from the class person and the images in the last column belong to none of the classes (called nobikenoperson). The second example in the last column shows a moped as a very difficult counter-example to the category of bikes.

N04). We are grateful to David Lowe and Cordelia Schmid for providing the code for their detectors/descriptors.

## References

1. S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proc. ECCV*, pages 113–130, 2002.
2. Gy. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *Proc. International Conference on Computer Vision*, 2003.
3. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.
4. W. Freeman and E. Adelson. The design and use of steerable filters. In *PAMI*, pages 891 – 906, 1991.
5. Y. Freund and R. E. Schapire. A decision-theoretic generalisation of on-line learning. *Computer and System Sciences*, 55(1), 1997.
6. A. Garg, S. Agarwal, and T. S. Huang. Fusion of global and local information for object detection. In *Proc. CVPR*, volume 2, pages 723–726, 2002.
7. R. C. Gonzalez and R.E. Woods. *Digital Image Processing.* Addison-Wesley, 2001.

8. L. Van Gool, T. Moons, and D. Ungureanu. Affine / photometric invariants for planar intensity patterns. In *Proc. ECCV*, pages 642 – 651, 1996.
9. C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. of the 4th ALVEY vision conference*, pages 147–151, 1988.
10. R. Laganiere. A morphological operator for corner detection. *Pattern Recognition*, 31(11):1643 – 1652, 1998.
11. T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 1996.
12. D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, 1999.
13. W. Maass and M. Warmuth. Efficient learning with virtual threshold gates. *Information and Computation*, 141(1):66–83, 1998.
14. S. Mahamud, M. Hebert, and J. Shi. Object recognition using boosted discriminants. In *Proc. CVPR*, volume 1, pages 551–558, 2001.
15. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. ICCV*, pages 525–531, 2001.
16. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*, pages 128–142, 2002.
17. K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *Proc. CVPR*, 2003.
18. C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. In *PAMI*, volume 19, pages 530–534, 1997.
19. C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, pages 151–172, 2000.
20. H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *International Journal of Computer Vision*, to appear.
21. E. Shilat, M. Werman, and Y. Gdalyahu. Ridge's corner detection and correspondence. In *Computer Vision and Pattern Recognition*, pages 976 – 981, 1997.
22. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, 2001.
23. M. Weber. *Unsupervised Learning of Models for Object Recognition.* PhD thesis, California Institute of Technology, Pasadena, CA, 2000.
24. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. ECCV*, 2000.
25. R. P. Wuertz and T. Lourens. Corner detection in color images by multiscale combination of end-stopped cortical cells. In *International Conference on Artificial Neuronal Networks*, pages 901 – 906, 1997.

# Object Level Grouping for Video Shots

Josef Sivic, Frederik Schaffalitzky, and Andrew Zisserman

Robotics Research Group, Department of Engineering Science,
University of Oxford
http://www.robots.ox.ac.uk/~vgg

**Abstract.** We describe a method for automatically associating image patches from frames of a movie shot into object-level groups. The method employs both the appearance and motion of the patches.
There are two areas of innovation: first, affine invariant regions are used to repair short gaps in individual tracks and also to join sets of tracks across occlusions (where many tracks are lost simultaneously); second, a robust affine factorization method is developed which is able to cope with motion degeneracy. This factorization is used to associate tracks into object-level groups.
The outcome is that separate parts of an object that are never visible simultaneously in a single frame are associated together. For example, the front and back of a car, or the front and side of a face. In turn this enables object-level matching and recognition throughout a video.
We illustrate the method for a number of shots from the feature film 'Groundhog Day'.

## 1 Introduction

The objective of this work is to automatically extract and group independently moving 3D semi-rigid (that is, rigid or slowly deforming) objects from video shots. The principal reason we are interested in this is that we wish to be able to match such objects throughout a video or feature length film. An object, such as a vehicle, may be seen from one aspect in a particular shot (e.g. the side of the vehicle) and from a different aspect (e.g. the front) in another shot. Our aim is to learn multi-aspect object models [19] from shots which cover several visual aspects, and thereby enable object level matching.

In a video or film shot the object of interest is usually tracked by the camera — think of a car being driven down a road, and the camera panning to follow it, or tracking with it. The fact that the camera motion follows the object motion has several beneficial effects for us: the background changes systematically, and may often be motion blurred (and so features are not detected there); and, the regions of the object are present in the frames of the shot for longer than other regions. Consequently, object level grouping can be achieved by determining the regions that are most common throughout the shot.

In more detail we define object level grouping as determining the set of appearance patches which (a) last for a significant number of frames, and (b) move (semi-rigidly) together throughout the shot. In particular (a) requires that every appearance of a patch is identified and linked, which in turn requires extended tracks for a patch – even associating patches across partial and complete occlusions. Such thoroughness has two benefits: first, the number of frames in which a patch appears really does correspond to the time that

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 85–98, 2004.
© Springer-Verlag Berlin Heidelberg 2004

it is visible in the shot, and so is a measure of its importance. Second, developing very long tracks significantly reduces the degeneracy problems which plague structure and motion estimation [5].

The innovation here is to use both motion and appearance consistency throughout the shot in order to group objects. The technology we employ to obtain appearance patches is that of affine co-variant regions [9,10,11,18]. These regions deform with viewpoint so that their pre-image corresponds to the same surface patch.

To achieve object level grouping we have developed the state of the art in two areas: first, the affine invariant tracked regions are used to repair short gaps in tracks (section 3) and also associate tracks when the object is partially or totally occluded for a period (section 5). The result is that regions are matched throughout the shot whenever they appear. Second, we develop a method of robust affine factorization (section 4) which is able to handle degenerate motions [17] in addition to the usual problems of missing and mis-matched points [1,3,7,13].

The task we carry out differs from that of layer extraction [16], or dominant motion detection where generally 2D planes are extracted, though we build on these approaches. Here the object may be 3D, and we pay attention to this, and also it may not always be the foreground layer as it can be partially or totally occluded for part of the sequence.

In section 6 we demonstrate that the automatically recovered object groupings are sufficient to support object level matching throughout the feature film 'Groundhog Day' [Ramis, 1993]. This naturally extends the frame based matching of 'Video Google' [14].

## 2 Basic Segmentation and Tracking

**Affine invariant regions.** Two types of affine invariant region detector are used: one based on interest point neighborhoods [10,11], the other based on the "Maximally Stable Extremal Regions" (MSER) approach of Matas *et al.* [9]. In both the detected region is represented by an ellipse. Implementation details of these two methods are given in the citations.

It is beneficial to have more than one type of region detector because in some imaged locations a particular type of feature may not occur at all. Here we have the benefit of region detectors firing at points where there is signal variation in more than one direction (e.g. near "blobs" or "corners"), as well as at high contrast extended regions. These two image areas are quite complementary. The union of both provides a good coverage of the image provided it is at least lightly textured, as can be seen in figure 1. The number of regions and coverage depends of course on the visual richness of the image.

To obtain tracks throughout a shot, regions are first detected independently in each frame. The tracking then proceeds sequentially, looking at only two consecutive frames at a time. The objective is to obtain correct matches between the frames which can then be extended to multi-frame tracks. It is here that we benefit significantly from the affine invariant regions: first, incorrect matches can be removed by requiring consistency with multiple view geometric relations: the robust estimation of these relations for point matches is very mature [6] and can be applied to the region centroids; second, the regions can be matched on their appearance. The latter is far more discriminating and invariant than the usual cross-correlation over a square window used in interest point trackers.

**Fig. 1. Example of affine invariant region detection.** (a) frame number 8226 from 'Groundhog Day'. (b) ellipses formed from 722 affine invariant interest points. (c) ellipses formed from 1269 MSER regions. Note the sheer number of regions detected just in a single frame, and also that the two types of region detectors fire at different and complementary image locations.

**Tracker implementation.** In a pair of consecutive frames, detected regions in the first frame are putatively matched with all detected regions in the second frame, within a generous disparity threshold of 50 pixels. Many of these putative matches will be wrong and an intensity correlation computed over the area of the elliptical region removes all putative matches with a normalized cross correlation below 0.90. The 1-parameter (rotation) ambiguity between regions is assumed to be close to zero, because there will be little cyclo-torsion between consecutive frames. All matches that are ambiguous, i.e. those that putatively match several features in the other frame, are eliminated.

Finally epipolar geometry is fitted between the two views using RANSAC with a generous inlier threshold of 3 pixels. This step is very effective in removing outlying matches whilst not eliminating the independent motions which occur between the two frames.

The results of this tracking on a shot from the movie 'Groundhog Day' are shown in figure 3b. This shot is used throughout the paper to illustrate the stages of the object level grouping. Note that the tracks have very few outliers.

## 3 Short Range Track Repair

The simple region tracker of the previous section can fail for a number of reasons most of which are common to all such feature trackers: (i) no region (feature) is detected in a frame – the region falls below some threshold of detection (e.g. due to motion blur); (ii) a region is detected but not matched due to a slightly different shape; and, (iii) partial or total occlusion.

The causes (i) and (ii) can be overcome by short range track repair using motion and appearance, and we discuss this now. Cause (iii) can be overcome by wide baseline matching on motion grouped objects within one shot, and discussion of this is postponed until section 5.

### 3.1 Track Repair by Region Propagation

The goal of the track repair is to improve tracking performance in cases where region detection or the first stage tracking fails. The method will be explained for the case of a one frame extension, the other short range cases (2-5 frames) are analogous.

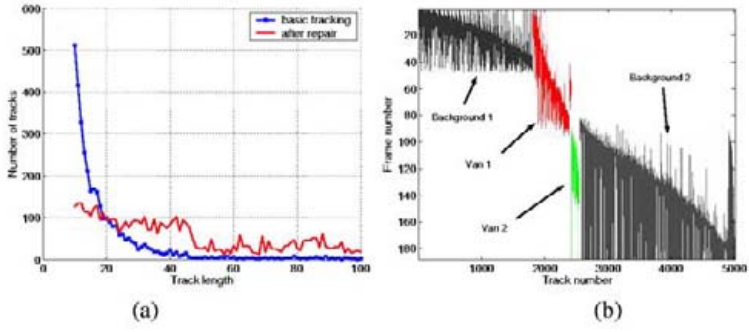



**Fig. 2.** (a) Histogram of track lengths for the shot shown in figure 3 for basic tracking (section 2) and after short range track repair (section 3). Note the improvement in track length after the repair – the weight of the histogram has shifted to the right from the mode at 10. (b) The sparsity pattern of the tracked features in the same shot. The tracks are coloured according to the independently moving objects, that they belong to, as described in section 4. The two gray blocks (track numbers 1-1808 and 2546-5011) correspond to the two background objects. The red and green blocks (1809-2415 and 2416-2545 respectively) correspond to van object before and after the occlusion.

The repair algorithm works on pairs of neighboring frames and attempts to extend already existing tracks which terminate in the current frame. Each region which has been successfully tracked for more than $n$ frames and for which the track terminates in the current frame is propagated to the next frame. The propagating transformation is estimated from a set of $k$ spatially neighboring tracks (here $n = 5$ and $k = 5$). In the case of successive frames only translational motion is estimated from the neighboring tracks. In the case of more separated frames the full affine transformation imposed by each tracked region should be employed.

It must now be decided if there is a detectable region near the propagated point, and if it matches an existing region. The refinement algorithm of Ferrari *et al.* [4] is used to fit the region locally in the new frame (this searches a hypercube in the 6D space of affine transformations by a sequence of line searches along each dimension). If the refined region correlates sufficiently with the original, then a new region is instantiated. It is here that the advantage of regions over interest points is manifest: this verification test takes account of local deformations due to viewpoint change, and is very reliable.

The standard 'book-keeping' cases then follow: (i) no new region is instantiated (e.g. the region may be occluded in the frame); (ii) a new region is instantiated, in which case the current track is extended; (iii) if the new instantiated region matches (correlates with) an existing region in its (5 pixel) neighborhood then this existing region is added to the track; (iv) if the matched region already belongs to a track starting in the new frame, then the two tracks are joined.

Figure 2 gives the 'before and after' histogram of track lengths, and the results of this repair are shown in figure 3. As can be seen, there is a dramatic improvement in the length of the tracks – as was the objective here. Note, the success of this method is due to the availability and use of two complementary constraints – motion and appearance.

**Fig. 3.** Example I: (a) 6 frames from one shot (188 frames long) of the movie 'Groundhog Day'. The camera is panning right, and the van moves independently. (b) frames with the basic region tracks superimposed. The tracked path $(x, y)$ position over time is shown together with each tracked region. (c) frames with short range repaired region tracks superimposed. Note the much longer tracks on the van after applying this repair. For presentation purposes, only tracks that last for more than 10 frames are shown. (d) One of the three dominant objects found in the shot. The other two are backgrounds at the beginning and end of the shot. No background is tracked in the middle part of the shot due to motion blur.

## 4 Object Extraction by Robust Sub-space Estimation

To achieve the final goal of identifying objects in a shot we must partition the tracks into groups with coherent motion. In other words, things that move together are assumed to belong together. For example, in the shot of figure 3 the ideal outcome would be the van as one object, and then several groupings of the background. The grouping constraint

used here is that of common (semi-)rigid motion, and we assume an affine camera model so the structure from motion problem reduces to linear subspace estimation.

For a 3-dimensional object, our objective would be to determine a 3D basis of trajectories $\mathbf{b}_k^i, k = 1, 2, 3$, (to span a rank 3 subspace) so that (after subtracting the centroid) all the trajectories $\mathbf{x}_j^i$ associated with the object could be written as [20]:

$$\mathbf{x}_j^i = \left(\mathbf{b}_1^i, \mathbf{b}_2^i, \mathbf{b}_3^i\right) (X_j, Y_j, Z_j)^\top$$

where $\mathbf{x}_j^i$ is the measured $(x, y)$ position of the $j$th point in frame $i$, and $(X_j, Y_j, Z_j)$ is the 3D affine structure.

The maximum likelihood estimate of the basis vectors and affine structure could then be obtained by minimizing the reprojection error

$$\sum_{ij} \left|\left| n_j^i \left(\mathbf{x}_j^i - \left(\mathbf{b}_1^i, \mathbf{b}_2^i, \mathbf{b}_3^i\right) (X_j, Y_j, Z_j)^\top\right)\right|\right|^2 \tag{1}$$

where $n_j^i$ is an indicator variable to label whether the point $j$ is (correctly) detected in frame $i$, and must also be estimated. This indicator variable is necessary to handle missing data.

It is well known [17] that directly fitting a rank 3 subspace to trajectories is often unsuccessful and suffers from over-fitting. For example, in a video shot the inter-frame motion is quite slow so using motion alone it easy to under-segment and group foreground objects with the background.

We build in immunity to this problem from the start, and fit subspaces in two stages: first, a low dimensional model (a projective homography) is used to hypothesize groups – this over-segments the tracks. These groups are then associated throughout the shot using track co-occurrences. The outcome is that trajectories are grouped into sets belonging to a single object. In the second stage 3D subspaces are robustly sampled from these sets, without over-fitting, and used to merge the sets arising from each object. These steps are described in the following sub-sections. This approach differs fundamentally from that of [1,3] where robustness is achieved by iteratively re-weighting outliers but no account is taken of motion degeneracy.

### 4.1 Basic Motion Grouping Using Homographies

To determine the motion-grouped tracks for a particular frame, both the previous and subsequent frames are considered. The aim is then to partition all tracks extending over the three frames into sets with a common motion. To achieve this, homographies are fitted to each pair of frames of the triplet using RANSAC [6], and an inlying set is scored by its error averaged over the three homographies. The inlying set is removed, and RANSAC is then applied to the remaining tracks to extract the next largest motion grouping, etc. This procedure is applied to every frame in the shot. This provides temporal coherence (since neighboring triplets share two frames) which is useful in the next step where motion groups are linked throughout the shot into an object.

### 4.2 Aggregating Segmentation over Multiple Frames

The problem with fitting motion models to pairs or triplets of frames are twofold: phantom motion cluster corresponding to a combination of two independent motions grouped

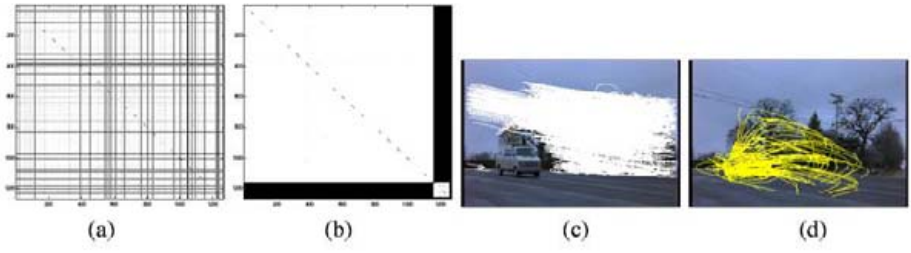



**Fig. 4. Aggregating segmentation over multiple frames**. (a) The track co-occurrence matrix for a ten frame block of the shot from figure 3. White indicates high co-occurrence. (b) The thresholded co-occurrence matrix re-ordered according to its connected components (see text). (c) (d) The sets of tracks corresponding to the two largest components (of size 1157 and 97). The other components correspond to 16 outliers.

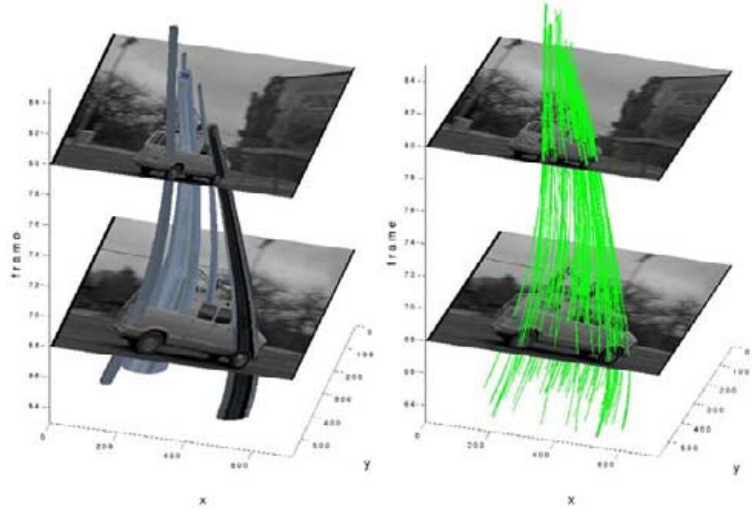



**Fig. 5. Trajectories following object level grouping**. Left: Five region tracks (out of a total 429 between these frames) shown as spatiotemporal "tubes" in the video volume. Right: A selection of 110 region tracks (of the 429) shown by their centroid motion. The frames shown are 68 and 80. Both figures clearly show the foreshortening as the car recedes into the distance towards the end of the shot. The number and quality of the tracks is evident: the tubes are approaching a dense epipolar image [2], but with explicit correspondence; the centroid motion demonstrates that outlier 'strands' have been entirely 'combed' out, to give a well conditioned track set.

together can ariser [15], and an outlying track will be occasionally, but not consistently, grouped together with the wrong motion group. In our experience these ambiguities tend not to be stable over many frames, but rather occasionally appear and disappear. To deal with these problems we devise a voting strategy which groups tracks that are consistently segmented together over multiple frames.

**Fig. 6.** Example II: object level grouping for another (35 frame) shot from the movie 'Groundhog Day'. Top row: The original frames of the shot. Middle and bottom row: The two dominant (measured by the number of tracks) objects detected in the shot. The number of tracks associated with each object is 721 (car) and 2485 (background).

The basic motion grouping of section 4.1 provides a track segmentation for each frame (computed using the two neighbouring frames too). To take advantage of temporal consistency the shot is divided into blocks of frames over a wider baseline of $n$ frames ($n = 10$ for example) and a track-to-track co-occurrence matrix $W$ is computed for each block. The element $w_{ij}$ of the matrix $W$ accumulates a vote for each frame where tracks $i$ and $j$ are grouped together. Votes are added for all frames in the block. In other words, the similarity score between two tracks is the number of frames (within the 10-frame block) in which the two tracks were grouped together. The task is now to segment the track voting matrix $W$ into temporally coherent clusters of tracks. This is achieved by finding connected components of a graph corresponding to the thresholded matrix $W$. To prevent under-segmentation the threshold is set to a value larger than half of the frame baseline of the block, i.e. 6 for the 10 frame block size. This guarantees that each track cannot be assigned to more than one group. Only components exceeding a certain minimal number of tracks are retained. Figure 4 shows an example of the voting scheme applied on a ten frame block from the shot of figure 3. This simple scheme segments the matrix $W$ reliably and overcomes the phantoms and outliers.

The motion clusters extracted in the neighbouring 10 frame blocks are then associated based on the common tracks between the blocks. The result is a set of connected clusters of tracks which correspond to independently moving objects throughout the shot.

### 4.3 Object Extraction

The previous track clustering step usually results in no more than 10 dominant (measured by the number of tracks) motion clusters larger than 100 tracks. The goal now is to identify those clusters that belong to the same moving 3D object. This is achieved by grouping pairs of track-clusters over a wider baseline of $m$ frames ($m > 20$ here). To test

**Fig. 7.** Example III: object level grouping for another (83 frame) shot from the movie Groundhog Day. Top row: The original frames of the shot. Middle and bottom row: The two dominant (measured by the number of tracks) objects detected in the shot. The number of tracks associated with each object is 225 (landlady) and 2764 (background). The landlady is an example of a slowly deforming object.

whether to group two clusters, tracks from both sets are pooled together and a RANSAC algorithm is applied to all tracks intersecting the $m$ frames. The algorithm robustly fits a rank 3 subspace as described in equation (1).

In each RANSAC iteration, four tracks are selected and full affine factorization is applied to estimate the three basis trajectories which span the three dimensional subspace of the ($2m$ dimensional) trajectory space. All other tracks that are visible in at least five views are projected onto the space. A threshold is set on reprojection error (measured in pixels) to determine the number of inliers. To prevent the grouping of inconsistent clusters a high number of inliers (90%) from both sets of tracks is required. When no more clusters can be paired, all remaining clusters are considered as separate objects.

### 4.4 Object Extraction Results

An example of one of the extracted objects (the van) is shown in figure 3d. In total, four objects are grouped for this shot, two corresponding to the van (before and after the occlusion by the post, see figure 9 in section 5) and two background objects at the beginning and end of the shot. The number of tracks associated with each object are 607 (van pre-occlusion), 130 (van post-occlusion), 1808 (background start) and 2466 (background end). The sparsity pattern of the tracks belonging to different objects is shown in figure 2(b). Each of the background objects is composed of only one motion cluster. The van object is composed of two motion clusters of size 580 and 27 which are joined at the object extraction RANSAC stage. The quality and coverage of the resulting tracks is visualized in the spatio-temporal domain in figure 5.

A second example of rigid object extraction from a different shot is given in figure 6. Figures 7 and 8 show examples of slowly deforming objects. This deformation is allowed

**Fig. 8.** Example IV: object level grouping for another (645 frame) shot from the movie Groundhog Day. Top row: The original frames of the shot where a person walks across the room while tracked by the camera. Middle and bottom row: The two dominant (measured by the number of tracks) objects detected in the shot. The number of tracks associated with each object is 401 (the walking person) and 15,053 (background). The object corresponding to the walking person is a join of three objects (of size 114, 146 and 141 tracks) connected by a long range repair using wide baseline matching, see figure 9b. The long range repair was necessary because the tracks are broken twice: once due to occlusion by a plant (visible in frames two and three in the first row) and the second time (not shown in the figure) due to the person turning his back on the camera. The trajectory of the regions is not shown here in order to make the clusters visible.

because rigidity is only applied over a sliding baseline of $m$ frames, with $m$ less than the total length of the track. For example we are able to track regions on slowly rotating and deforming face such as a mouth opening.

## 5 Long Range Track Repair

The object extraction method described in the previous section groups objects which are temporally coherent. The aim now is to connect objects that appear several times throughout a shot, for example an object that disappears for a while due to occlusion. Typically a set of tracks will terminate simultaneously (at the occlusion), and another set will start (after the occlusion). The situation is like joining up a cable (of multiple tracks) that has been cut.

The set of tracks is joined by applying standard wide baseline matching [9,11,18] to a pair of frames that each contain the object. There are two stages: first, epsilon-nearest neighbor search on a SIFT descriptor [8] for each region, is performed to get a set of putative region matches, and second, this set is disambiguated by a local spatial consistency constraint: a putative match is discarded if it does not have a supporting match within its k-nearest spatial neighbors [12,14]. Since each region considered for matching is part of a track, it is straightforward to extend the matching to tracks. The two objects

**Fig. 9.** Two examples of long range repair on (a) shot from figure 3 where a van is occluded (by a post) which causes the tracking and motion segmentation to fail, and (b) shot from figure 8 where a person walks behind a plant. First row: Sample frames from the two sequences. Second row: Wide-baseline matches on regions of the two frames. The green lines show links between the matched regions. Third row: Region tracks on the two objects that have been matched in the shot.

are deemed matched if the number of matched tracks exceeds a threshold. Figure 9 gives two examples of long range repair on shots where the object was temporarily occluded.

## 6 Application: Object Level Video Matching

Having computed object level groupings for shots throughout the film, we are now in a position to retrieve object matches given only part of the object as a query region. Grouped objects are represented by the union of the regions associated with all of the object's tracks. This provides an implicit representation of the 3D structure, and is sufficient for matching when different parts of the object are seen in different frames. In more detail, an object is represented by the set of regions associated with it in each key-frame. As shown in figures 10 and 11, the set of key-frames naturally spans the object's visual aspects contained within the shot.

In the application we have engineered, the user outlines a query region of a key-frame in order to obtain other key-frames or shots containing the scene or object delineated by the region. The objective is to retrieve *all* key-frames/shots within the film containing the object, even though it may be imaged from a different visual aspect.

The object-level matching is carried out by determining the set of affine invariant regions enclosed by the query region. The convex hull of these tracked regions is then computed in each key frame, and this hull determines in turn a query region for that frame. Matching is then carried out for all query regions using the Video Google method described in [14].

An example of object-level matching throughout a database of 5,641 key-frames of the entire movie 'Groundhog Day' is shown in figures 10 and 11.

**Fig. 10.** Object level video matching I. Top row: The query frame with query region (side of the van) selected by the user. Second row: The automatically associated keyframes and outlined query regions. Next four rows: Example frames retrieved from the entire movie 'Groundhog Day' by the object level query. Note that views of the van from the back and front are retrieved. This is not possible with wide-baseline matching methods alone using only the side of the van visible in the query image.

## 7 Discussion and Extensions

We have shown that representing an object as a set of viewpoint invariant patches has a number of beneficial consequences: gaps in tracks can be reliably repaired; tracked objects can be matched across occlusions; and, most importantly here, different viewpoints of the object can be associated provided they are sampled by the motion within a shot.

**Fig. 11.** Object level video matching II. Top row: The query frame with query region selected by the user. The query frame acts as a portal to the keyframes associated with the object by the motion-based grouping (shown in the second row). Note that in the associated keyframes the person is visible from the front and also changes scale. See figure 8 for the corresponding object segmentation. Next three rows: Example frames retrieved from the entire movie 'Groundhog Day' by the object level query.

We are now at a point where useful object level groupings can be computed automatically for shots that contain a few objects moving independently and semi-rigidly. This has opened up the possibility of pre-computing object-level matches throughout a film – so that content-based retrieval for images can access objects directly, rather than image regions; and queries can be posed at the object, rather than image, level.

**Acknowledgements.** We are very grateful to Jiri Matas and Jan Paleček for their MSE region detector. Shots were provided by Mihai Osian from KU Leuven. This work was supported by EC project Vibes and Balliol College, Oxford.

## References


1. H. Aanaes, R. Fisker, K. Astrom, and J. M. Carstensen. Robust factorization. *IEEE PAMI*, 24:1215–1225, 2002.
2. R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *IJCV*, 1(1):7–56, 1987.
3. F. De la Torre and M. J. Black. A framework for robust subspace learning. *IJCV*, 54:117–142, 2003.
4. V. Ferrari, T. Tuytelaars, and L. Van Gool. Wide-baseline multiple-view correspondences. In *Proc. CVPR*, pages 718–725, 2003.
5. A. Fitzgibbon and A. Zisserman. Automatic camera tracking. In Shah and Kumar, editors, *Video Registration*. Kluwer, 2003.
6. R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
7. D. W. Jacobs. Linear fitting with missing data: applications to structure-from-motion and to characterizing intensity images. In *Proc. CVPR*, pages 206–212, 1997.
8. D. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, pages 1150–1157, 1999.
9. J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. BMVC.*, pages 384–393, 2002.
10. K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proc. ECCV*. Springer-Verlag, 2002.
11. F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Proc. ECCV*, volume 1, pages 414–431. Springer-Verlag, 2002.
12. C. Schmid. *Appariement d'Images par Invariants Locaux de Niveaux de Gris*. PhD thesis, L'Institut National Polytechnique de Grenoble, Grenoble, 1997.
13. H.-Y. Shum, I. Ikeuchi, and R. Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE PAMI*, 17:854–867, 1995.
14. J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc. ICCV*, 2003.
15. P. H. S. Torr. *Motion segmentation and outlier detection*. PhD thesis, Dept. of Engineering Science, University of Oxford, 1995.
16. P. H. S. Torr, R. Szeliski, and P. Anadan. An integrated bayesian approach to layer extraction from image sequence. *IEEE PAMI*, 23:297–304, 2001.
17. P. H. S. Torr, A. Zisserman, and S. Maybank. Robust detection of degenerate configurations for the fundamental matrix. *CVIU*, 71(3):312–333, 1998.
18. T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. BMVC.*, pages 412–425, 2000.
19. C. Wallraven and H. Bulthoff. Automatic acquisition of exemplar-based representations for recognition from image sequences. In *CVPR Workshop on Models vs. Exemplars*, 2001.
20. L. Zelnik-Manor and M. Irani. Multi-view subspace constraints on homographies. In *Proc. ICCV*, 1999.

# Statistical Symmetric Shape from Shading for 3D Structure Recovery of Faces

Roman Dovgard and Ronen Basri

Dept. of Applied Mathematics and Computer Science,
Weizmann Institute of Science,
Rehovot 76100, Israel
{romad,ronen}@wisdom.weizmann.ac.il

**Abstract.** In this paper, we aim to recover the 3D shape of a human face using a single image. We use a combination of symmetric shape from shading by Zhao and Chellappa and statistical approach for facial shape reconstruction by Atick, Griffin and Redlich. Given a single frontal image of a human face under a known directional illumination from a side, we represent the solution as a linear combination of basis shapes and recover the coefficients using a symmetry constraint on a facial shape and albedo. By solving a single least-squares system of equations, our algorithm provides a closed-form solution which satisfies both symmetry and statistical constraints in the best possible way. Our procedure takes only a few seconds, accounts for varying facial albedo, and is simpler than the previous methods. In the special case of horizontal illuminant direction, our algorithm runs even as fast as matrix-vector multiplication.

## 1 Introduction

The problem of estimating the shape of an object from its shading was first introduced by Horn [1]. He defined the mapping between the shading and surface shape in terms of the reflectance function $I_{x,y} = R(p,q)$ where $I_{x,y}$ denotes image intensity, $p \doteq z_x$ and $q \doteq z_y$, $z$ being the depth of the object and $(x,y)$ are projected spatial coordinates of the 3D object. In this paper we will assume orthographic projection and Lambertian reflectance model, thus obtaining the following *brightness constraint*:

$$I_{x,y} \propto \rho_{x,y} \frac{1 - pl - qk}{\sqrt{p^2+q^2+1}\sqrt{l^2+k^2+1}}, \tag{1}$$

where $\frac{(l,k,1)}{||(l,k,1)||}$ is the illuminant direction (we have here proportion, instead of equality, because of the light source intensity). The task of a shape from shading algorithm is to estimate the unknowns of Eq. (1), which are the surface albedos $\rho_{x,y}$, and the surface depths $z_{x,y}$. With only image intensities known, estimating both the depths and the albedos is ill-posed. A common practice is to assume a constant surface albedo, but in a survey [2] it is concluded that depth estimates for real images come out to be very poor with this simplistic assumption.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 99–113, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this paper, we aim to recover the 3D shape of a human face using a single image. In this case there are some other constraints, that can be imposed on the unknown variables $\rho_{x,y}$ and $z_{x,y}$ in Eq. (1). Shimshoni et al. [3] and Zhao and Chelloppa [4,5] presented shape from shading approaches for symmetric objects, which are applicable to human faces. In [3], geometric and photometric stereo are combined to obtain a 3D reconstruction of quasi-frontal face images. In [4,5] *symmetry constraints* on depth and albedo are used to obtain another, *albedo-free brightness constraint*. Using this constraint and Eq. (1) they find a depth value at every point. Lee and Rosenfeld [6] presented an approximate albedo estimation method for scene segmentation. Using this albedo estimation method, Tsai and Shah [7] presented a shape from shading algorithm, which starts with segmenting a scene into a piece-wise constant albedo regions, and then applies some shape from shading algorithm on each region independently. Nandy and Ben-Arie [8,9] assume constant facial albedo and recover facial depth using neural networks. All the methods mentioned above assume either constant or piece-wise constant albedo, which is not a realistic assumption for real 3D objects such as human faces. Finally, Ononye and Smith [10] used color images for 3D recovery and obtained good results for simple objects. Results which they showed on *unrestricted* facial images are not so accurate, however, probably because of the lack of independence of the $R, G, B$ channels in these images.

Atick et al. [11] and Vetter et al. [12,13,14,15] provided statistical shape from shading algorithms, which attempt to reconstruct the 3D shape of a human face using statistical knowledge about the shapes and albedos of human faces in general. In [11], a constant facial albedo is assumed and a linear constraint on the shape is imposed. The authors of [12,13,14,15], went a step further and have dropped the constant albedo assumption, imposing linear constraint on both texture (albedo map) and shape of the face. Because facial texture is not as smooth in general as the facial shape, imposing linear constraint on the texture requires a special preprocessing stage to align the database facial images to better match each other. Both approaches use certain optimization methods to find the coefficients of the linear combinations present in the linear constraints they are using, thus providing no closed-form solution, and consuming significant computational time.

We present an algorithm which accounts for varying facial albedo. Our method provides a closed-form solution to the problem by solving a single least-squares system of equations, obtained by combining *albedo-free brightness* and *class linearity* constraints. Our approach requires a restrictive setup: frontal face view, known directional illumination (that can be estimated for example by Pentland's method [16]) and Lambertian assumption about the face. We also get some inaccuracies in the reconstructed faces because human faces are not perfectly symmetric [17]. However, this is the first algorithm for 3D face reconstruction from a single image, which provides a closed-form solution within a few seconds.

The organization of the paper is as follows. In Sect. 2, we describe in detail previous work that is relevant to our approach, mainly symmetric and statistical

shape from shading algorithms. Later, we present our algorithm in Sect. 3, and give experimental results in Sect. 4. Finally, we draw conclusions in Sect. 5.

## 2 Previous Work

### 2.1 Symmetric Shape from Shading

Zhao and Chellappa [4,5] introduced symmetric shape from shading algorithm which uses the following *symmetry constraint*:

$$\rho_{x,y} = \rho_{-x,y}, \quad z_{x,y} = z_{-x,y} \tag{2}$$

to recover the shape of bilaterally symmetric objects, e.g. human faces. From *brightness* and *symmetry constraints* together the following equations follow:

$$I_{-x,y} \propto \rho_{x,y} \frac{1 + pl - qk}{\sqrt{p^2 + q^2 + 1}\sqrt{l^2 + k^2 + 1}} \tag{3}$$

$$\frac{I_{-x,y} - I_{x,y}}{I_{-x,y} + I_{x,y}} = \frac{pl}{1 - qk} \tag{4}$$

Denoting $D_{x,y} \doteq I_{-x,y} - I_{x,y}$ and $S_{x,y} \doteq I_{-x,y} + I_{x,y}$ we obtain the following *albedo free brightness constraint*:

$$Slp + Dkq = D. \tag{5}$$

Using Eq. (5), Zhao and Chellappa write $p$ as a function of $q$, and substitute it into Eq. (1), obtaining equation in two unknowns, $q$ and the albedo. They approximate the albedo by a piece-wise constant function and solve the equation for $q$. After recovering $q$ and then $p$, the surface depth $z$ can be recovered by any surface integration method, e.g., the one used in [18].

Yilmaz and Shah [19] tried to abandon the albedo piece-wise constancy assumption by solving Eq. (5) directly. They wrote Eq. (5) as an equation in $z_{x,y}$, instead of $p$ and $q$, and tried to solve it iteratively. A linear partial differential equation

$$a_{x,y} z_x + b_{x,y} z_y = c_{x,y}, \tag{6}$$

of the same type in $z$, in different context, appears in the linear shape from shading method of Pentland [20], and was used also in [21].

Pentland tried to solve Eq. (6) by taking the Fourier transform of both sides, obtaining

$$A_{u,v}(-iu) Z_{u,v} + B_{u,v}(-iv) Z_{u,v} = C_{u,v}, \tag{7}$$

where $A, B, C$ and $Z$ stand for the Fourier transforms of $a, b, c$ and $z$, respectively. It was stated in [20], that $Z_{u,v}$ can be computed by rearranging terms in Eq. (7) and taking the inverse Fourier transform. However, rearranging terms in Eq. (7) results in $Z_{u,v} = iC_{u,v}/(A_{u,v}u + B_{u,v}v)$. This equation is undefined when $A_{u,v}u + B_{u,v}v$ vanishes, and thus it leaves ambiguities in $Z_{u,v}$, and therefore also in $z_{x,y}$.

As was noted in [22], but was noticed neither in [19] nor in [20,21], Eq. (6), being a linear partial differential equation in $z$, can be solved only up to a *one dimensional ambiguity* (hidden in the initial conditions), for example via the characteristic strip method [23]. Hence, Eq. (5) cannot be solved alone, without leaving large ambiguities in the solution.

### 2.2 Statistical Shape from Shading

Atick et al. [11] took a collection of 200 heads where each head was scanned using a CyberWare Laser scanner. Each surface was represented in cylindrical coordinates as $r^t(\theta, h), 1 \leq t \leq 200$ with 512 units of resolution for $\theta$ and 256 units of resolution for $h$. After cropping the data to a $256 \times 200$ (angular $\times$ height) grid around the nose, it was used to build *eigenhead decomposition* as explained below. They took $r_0(\theta, h)$ to be the average of 200 heads and then performed principal component analysis [24] obtaining the eigenfunctions $\Psi_i$, such that any other human head could be represented by a linear combination $r(\theta, h) = r_0(\theta, h) + \sum_{i=1}^{199} \beta_i \Psi_i(\theta, h)$. After that they applied conjugate gradient optimization method to find the coefficients $\beta_i$, such that the resulting $r(\theta, h)$ will satisfy the brightness constraint in Eq. (1), assuming constant albedo.

In [12,13,14,15], the authors went a step further and have dropped the constant albedo assumption, imposing linear constraint on both texture (albedo map) and shape of the face. In order to achieve shape and texture coordinate alignment between the basis faces, they parameterized each basis face by a fixed collection of 3D vertices $(X_j, Y_j, Z_j)$, called a point distribution model, with associated color $(R_j, B_j, G_j)$ for each vertex. Enumeration of the vertices was the same for each basis face. They modelled the texture of a human face by eigenfaces [25] and its shape by eigenheads, as described above. They recovered sets of texture and shape coefficients via complex multi-scale optimization techniques. These papers also treated non-Lambertian reflectance, which we do not treat here.

Both statistical shape from shading approaches described above do not use the simple Cartesian $(x, y) \mapsto z(x, y)$ parameterization for the eigenhead surfaces. While the parameterizations described above are more appropriate for capturing linear relations among the basis heads, they have the drawback of projecting the same head vertex onto different image locations, depending on the shape coefficients. Thus, image intensity depends on the shape coefficients. To make this dependence linear, the authors in [11] used Taylor expansion of the image $I$, with approximation error consequences. In [12,13,14,15], this dependence is taken into account in each iteration, thus slowing down the convergence speed.

## 3 Statistical Symmetric Shape from Shading

We use in this paper a database of 138 human heads from the USF Human-ID 3D Face Database [26] scanned with a CyberWare Laser scanner. Every head,

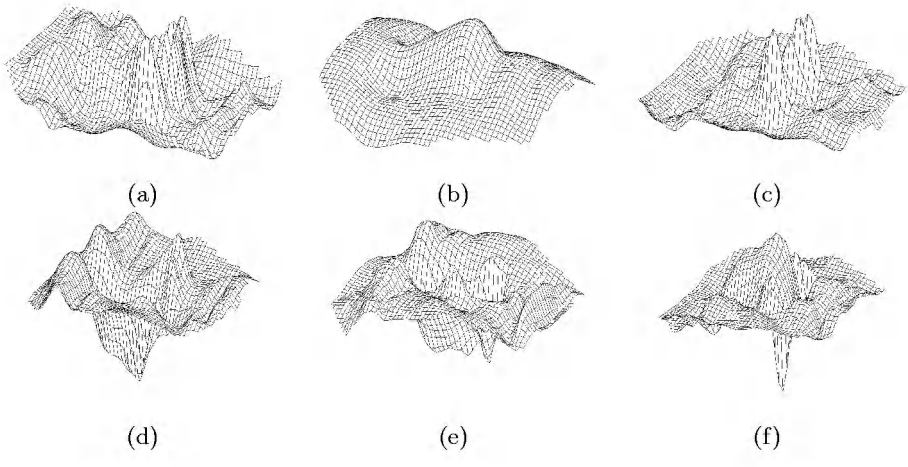



**Fig. 1.** (a) Standard deviation surface $z^{\mathtt{std}}$. Note high values of $z^{\mathtt{std}}$ around the nose for example: this is problem of Cartesian representation - high $z$ variance in areas with high spatial variance, like area around the nose. (b)-(f) The five most significant eigenheads $z^1, \ldots, z^5$, out of the 130 eigenheads.

originally represented in the database in cylindrical coordinates, is resampled to a Cartesian grid of resolution $142 \times 125$. Then, we have used a threshold on the standard deviation $z^{\mathtt{std}}$ of the facial depths, in order to mask out pixels $(x, y)$ which are not present in all the basis faces, or alternatively are unreliable (see Fig. 1 (a) for the masked standard deviation map). After that we perform PCA on the first 130 heads, obtaining the 130 eigenheads $z^i$ (first five of which are shown in Fig. 1 (b)-(f)), and keeping the remaining eight heads for testing purposes. We then constrain the shape of a face to be reconstructed, $z_{x,y}$, to the form:

$$z_{x,y} = \sum_{i=1}^{130} \alpha_i z^i_{x,y}, \tag{8}$$

for some choice of coefficients $\{\alpha_i\}$.

Since our *face space constraint* (8) is written in a Cartesian form, we can take derivatives w.r.t. $x$ and $y$ of both sides to obtain face space constraints on $p, q$:

$$p = \sum_{i=1}^{130} \alpha_i p^i, \quad q = \sum_{i=1}^{130} \alpha_i q^i, \tag{9}$$

where $p^i \doteq z^i_x$ and $q^i \doteq z^i_y$. The two equations above, together with the albedo free brightness constraint (5) result in the following equation chain:

$$D = Slp + Dkq = Sl \sum_{i=1}^{130} \alpha_i p^i + Dk \sum_{i=1}^{130} \alpha_i q^i = \sum_{i=1}^{130} (Slp^i + Dkq^i)\alpha_i. \tag{10}$$

This equation is linear in the only unknowns $\alpha_i, 1 \leq i \leq 130$. We find a least-squares solution, and then recover $z$ using Eq. (8). One can speed up calculations by using less than 130 eigenheads in the face space constraint (8).

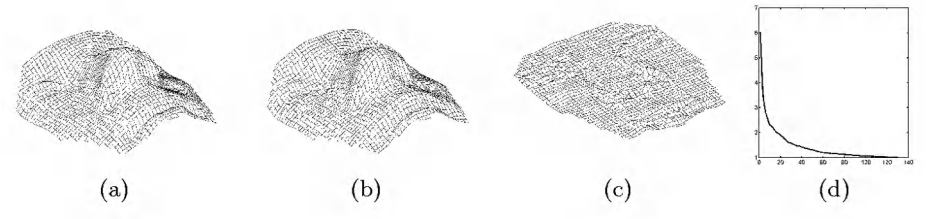



**Fig. 2.** Testing of *generalization* ability of our eigenhead decomposition. (a) One of the eight out-of-sample heads not used to construct the face space. (b) Projection of surface in (a) onto the 130-dimensional face space. (c) Error surface. Note some correlation with standard deviation surface from Fig. 1. (d) Plot of dependance of the reconstruction error on the number of modes used in the representation. The error is defined as $|z^{\texttt{actual}} - z^{\texttt{estimated}}| \cdot \cos\sigma^{\texttt{slant}}(z^{\texttt{actual}})$ averaged over all points on the surface and over eight out-of-sample heads, and is displayed as a percentage from the whole dynamic range of $z$. Error is normalized by the cosine of the slant angle, to account for the fact that the actual distance between two surfaces $z^{\texttt{actual}}$ and $z^{\texttt{estimated}}$ is approximately the distance along the $z$-axis times the cosine of the slant angle of the ground truth surface.

The choice of Cartesian coordinates was necessary for obtaining face space constraints on $p$ and $q$ in Eq. (9). Although Cartesian parameterization is less appropriate for eigenhead decomposition than cylindrical or point distribution model parameterizations, it provides eigenhead decomposition with sufficient *generalization* ability as is shown in Fig. 2 - in reconstruction example (a)-(c) we see relatively unnoticeable error, and in (d) we see a rather fast decay of the *generalization error* when the number of basis heads increases. Dividing the generalization error with the first eigenhead only (which is just the average head) by the generalization error with 130 eigenheads we obtain the *generalization quality* $\frac{||z^{\texttt{actual}} - z^m||}{||z^{\texttt{actual}} - z^{\texttt{estimated}}||} = 5.97$ (here $z^m$ stands for the average of the first 130 heads, and $z^{\texttt{actual}}$ with $z^{\texttt{estimated}}$ stand for the true and estimated depths, respectively), while [11] achieves with cylindrical coordinates a generalization quality of about 10. We provide in Sect. 3.3 a recipe for improving the generalization ability of the model, albeit without giving empirical evidence for it. The generalization errors we have right now are insignificant, comparing to errors in the reconstructions themselves, so that they do not play a major role in the accuracy of the results.

### 3.1 Special Case of $k = 0$

In the case of $k = 0$, Eq. (10) has a simpler form,

$$\frac{D}{Sl} = \sum_{i=1}^{130} p^i \alpha_i. \tag{11}$$

Setting $\{p^i\}$ to be the principal components of the $x$ derivatives of the facial depths, and $z^i$ accordingly to be the linear combinations of the original depths with the same coefficients which are used in $p^i$, one could solve for $\alpha$'s by setting $\alpha_i = \left\langle \frac{D}{Sl}, p^i \right\rangle$ (this is $\approx 150$ times faster than in our general method; in fact, this is as fast as a matrix-vector multiplication), and for $z$ via Eq. (8). This could be done up to a scaling factor, even without knowledge of light source direction.

Because different heads, which were used to build the face space, have face parts with different $y$ coordinates, the face space has some vertical ambiguities. This means that for a given face in the face space, also some face with $y$ shifts of a face components is also in the face space. In contrast to the general case, where ambiguity is driven by PDE's characteristic curves, and therefore is pretty random, in the case of $k = 0$ ambiguity is in vertical direction and thus is going in resonance with ambiguity in the face space. So, in the case of $k = 0$, final solution also will contain vertical ambiguities of the type $z_{x,y} = z^0_{x,y} + A(y)$ (where $A(y)$ is the ambiguity itself). We will show empirically, that due to vertical ambiguities in the the face space, results turn out to be quite inaccurate in this special case.

### 3.2 Extending Solution's Spatial Support

All basis heads used in build-up of the face space have different $x, y$ support, and spatial support of the eigenheads is basically limited to the intersection of their supports. Therefore spatial support of the reconstructed face area is also limited. To overcome this shortcoming of our algorithm, one can fit surface parameterized via a point distribution model (PDM) [12,13,14,15] to our solution surface $z = z_{x,y}$. This fitting uses our partial solution surface for 3D reconstruction of the whole face, and is straightforward, as opposed to fitting of PDM to a 2D image.

### 3.3 Improving Generalization Ability

In the Cartesian version of the face space (8) eigenheads do not match each other perfectly. For example noses of different people have different sizes in both $x, y$ and $z$ directions. A linear combination of two noses with different $x, y$ support produces something which is not a nose. Suppose now that we have a certain face with certain $x, y$ support for its nose. In order to get this nose we need basis faces with noses of similar $x, y$ support to this nose. This means that only a few basic faces will be used in a linear combination to produce this particular nose. This observation explains why a Cartesian version of face space has the highest generalization error among all face space representations: other representations have a better ability to match the supports of different face parts such as noses.

In order to overcome the drawbacks mentioned above, of the Cartesian principle component decomposition, we suggest another Cartesian decomposition for the shape space of human faces. Using NMF [27] or ICA [28,29], it is possible to decompose the shape of a human face as a sum of its basic parts $z^i, 1 \leq i \leq 130$. In contrast to the principal component analysis, each $z^i$ will have a compact $x, y$ support. Therefore, from each $z^i$ several $z^i_j$ can be derived, which are just slightly shifted and/or scaled copies of the original $z^i$, in the $x, y$ plane, in random directions. Using these shifted (and/or scaled) copies of original $z^i$, $1 \leq i \leq 130$, a broader class of face shapes can be obtained by a linear combination.

An alternative method for improving the generalization quality of the face space is to perform a spatial alignment of DB faces via image warping using for example manually selected landmarks [30]. For a progressive alignment it is possible to use eigenfeatures, which can be computed from the aligned DB faces.

## 4 Experiments

We have tested our algorithm on the Yale face database B [31]. This database contains frontal images of ten people from varying illumination angles. Alignment in the $x, y$ plane between the faces in the Yale database and the 3D models is achieved using spatial coordinates of the centers of the eyes. We use eye coordinates of people from the Yale database, which are available. Also we marked the eye centers for our $z^m$ and used it for the alignment with the images.

For every person in the Yale face database B, there are 63 frontal facial views with different known point illuminations stored. Also for each person its ambient image is stored. As in this paper we do not deal with the ambient component we subtract the ambient image from each one of the 63 images, prior to using them. As we need ground truth depths for these faces for testing purposes, and it is not given via a laser scanner for example, we first tried to apply photometric stereo on these images, in order to compute the depths. However, because the 63 light sources have different unknown intensities, performing photometric stereo is impossible. Hence, we have taken a different strategy for "ground truth" depth computation.

We took two images, taken under different illumination conditions, for each one of the ten faces present in the database. A frontal image $F$ with zero azimuth $Az$ and elevation $El$, and image $I$ with azimuth angle $Az = 20°$ and elevation angle $El = 10°$. Substituting $l = -\tan 20°$ and $k = \tan 10°/\cos 20°$ into Eq. (1), we obtain a Lambertian equation for the image $I$. Doing the same with $l = 0$ and $k = 0$ we obtain a Lambertian equation for the image $F$. Dividing these two equations, we obtain the following equation, with $l = -\tan 20°$ and $k = \tan 10°/\cos 20°$ (we have here equality up to scaling factor $\lambda$ caused by a difference of light source intensities used to produce the images $I$ and $F$):

$$\lambda \frac{I_{x,y}}{F_{x,y}} = 1 - pl - qk. \tag{12}$$

**Table 1.** Quality and computational complexity of our algorithm, applied to all the ten Yale faces. Ten columns correspond to the ten faces. The first row contains asymmetry estimates of the faces (we subtract the lowest element). The second row contains the quality estimates, measured via an inverse normalized distance between the estimated and actual depths. The third row contains the quality estimates with statistical Cartesian shape from shading (assuming constant albedo as in the work of Atick et al. [11]). The last row contains running times (in seconds) of our algorithm on all ten faces

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Asymmetry | 0.53 | 1.08 | 0 | 0.52 | 1.36 | 0.87 | 0.75 | 0.68 | 0.65 | 0.38 |
| Quality | 1.81 | 1.78 | 3.39 | 1.92 | 1.91 | 1.29 | 1.96 | 2.59 | 1.45 | 2.94 |
| Const Albedo | 1.27 | 1.24 | 1.76 | 1.41 | 1.57 | 1.10 | 2.15 | 2.5 | 1.5 | 2.35 |
| Running Time | 1.76 | 1.76 | 1.76 | 1.76 | 2.21 | 1.76 | 1.75 | 1.76 | 1.75 | 1.76 |

Eqs. (12) and (9) together enable us to get the "ground truth" depth, given the scaling factor $\lambda$ (which we will show how to calculate, later on):

$$F_{x,y} - \lambda I_{x,y} = \sum_{i=1}^{130} (F_{x,y} l p^i + F_{x,y} k q^i) \alpha_i. \tag{13}$$

Of coarse such a "ground truth" is less accurate than one obtained with photometric stereo, because it is based on two, rather than on all 63 images, and because it uses the face space decomposition, which introduces its own generalization error. Also small differences between the images $I$ and $F$ cause small errors in the resulting "ground truth". Still this "ground truth" has major advantage over results obtained by our symmetric shape from shading algorithm, which is that it does not use an inaccurate symmetry assumption about the faces [17].

Our algorithm estimates the depths of each one of the ten faces by solving Eq. (10). Then we take the estimated $\alpha_i$'s and plug them into Eq. (13). We find the best $\lambda$ satisfying this equation and use it in the "ground truth" calculation. This mini-algorithm for $\lambda$ calculation is based on the fact that our estimation, and the ground truth are supposed to be close, and therefore our estimation can be used to reduce ambiguity in the "ground truth" solution. As $\lambda$ has some error, we need to perform a small additional alignment between the "ground truth" and the estimated depths. We scale the "ground truth" solution at the end, so that it will have the same mean as the estimated depth.

In Table 1 we provide asymmetry estimates for all ten Yale DB faces along with quality and computational complexity estimates of the results. Asymmetry is measured via a Frobenius distance between normalized (to mean gray level 1) frontally illuminated face $F$ and its reflect $R$. Quality is measured by the fraction $||z^{\texttt{actual}} - z^m||/||z^{\texttt{actual}} - z^{\texttt{estimated}}||$, and computational complexity measured by a running time, in MATLAB, on a Pentium 4 1600MHz computer. Correlation coefficient between the facial asymmetry and resulting quality estimates is -0.65, indicating a relatively strong anti-correlation between quality and asymmetry.

**Table 2.** Quality of results of our algorithm for the case that $k = 0$, on all the ten Yale faces. The ten columns correspond to the ten faces. We performed quality estimation for reconstructions from images with $Az = -10^\circ$ and $Az = -25^\circ$. Prior to estimating the quality, we shifted and stretched the estimated depths, so that they will have the same mean and variance as the "ground truth" depths. We estimated quality with and without, row normalization, in which each depth row to be estimated was normalized to have the same mean as its counterpart row in the "ground truth" depth

| Without normalization (true quality) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $Az = -10^\circ$ | 0.70 | 0.96 | 0.88 | 1.25 | 0.76 | 0.90 | 1.06 | 1.63 | 0.81 | 1.20 |
| $Az = -25^\circ$ | 0.86 | 1.24 | 1.14 | 1.21 | 0.91 | 0.68 | 1.29 | 1.81 | 0.92 | 1.50 |

| With normalization (additional test) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $Az = -10^\circ$ | 1.34 | 1.63 | 1.58 | 1.69 | 1.52 | 1.11 | 1.46 | 2.92 | 1.54 | 2.07 |
| $Az = -25^\circ$ | 1.70 | 2.20 | 2.27 | 1.93 | 2.12 | 0.90 | 2.08 | 3.14 | 1.88 | 2.27 |

Results with Cartesian face space, replacing the symmetry with the constant albedo assumption (thus simulating the work of Atick et al. [11]), are found via iterative optimization on the $\alpha_i$'s according to Eq. (1), initialized by our solution. For each database image we chose its best scale to fit into the Lambertian equation, thus making the statistical shape from shading results as good as possible for a faithful comparison. Our results are slightly better on most faces (Table 1).

The three best results of our algorithm, with quality at least 2.5 (faces 3,8 and 10 in the Yale ordering), are depicted in Fig. 3 (along with their statistical SFS counterparts). Also, in the first three rows of Fig. 4, we show textured faces (with texture being the image of frontally illuminated face), rendered with our "ground truth", estimated and average depths. In the first three rows of Fig. 5, we render these faces as if they were shot using frontal illumination, by taking images with $Az = 20^\circ$ and $El = 10^\circ$ and cancelling out side illumination effect by dividing them by $1 - lp - kq$, where $p$ and $q$ are recovered by our algorithm from the image $I$ and are given directly (without using $z$) by Eq. (9).

In the last row of Fig. 4, we show results for the face number 6 in the Yale database, which has the worst reconstruction quality (see Table 1). In the last row of Fig. 5, we show renderings for this face. One can note a significant asymmetry of the face, which explains the rather bad reconstruction results in Fig. 4. We provide, in the additional material, results of the algorithm on *all* ten Yale faces.

We attribute inaccurate results of our algorithm, on many faces, to facial asymmetry. Results in Fig. 5 can be compared with similar results by Zhao and Chellappa [4,5] (see Figs. 14 and 15 in [4]). Note that both are affected by facial asymmetry. Using some illumination invariant feature matcher [32], features on two sides of a face, could be matched based solely on the albedo, and warp the face to one with symmetric shape and texture, but a warped illumination (with less impact on errors, due to smoothness of the illumination). However, we doubt whether this approach is feasible, because of the matching errors.

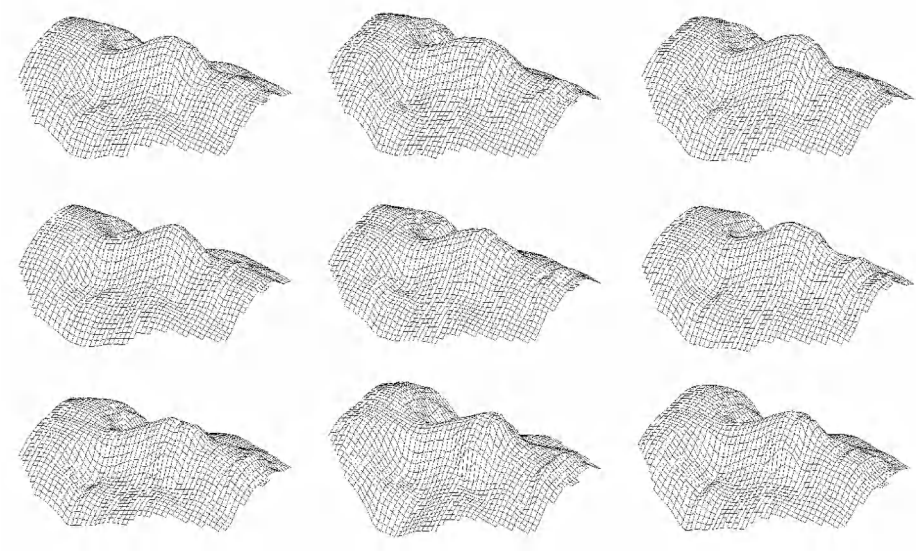


**Fig. 3.** Three columns correspond to three different faces from the Yale face database B. First row contains meshes of the faces with "ground truth" depths. Second row contains the meshes reconstructed by our algorithm from images with lighting $Az = 20^\circ$ and $El = 10^\circ$. Third row contains reconstructions from statistical Cartesian SFS algorithm.

### 4.1 Results in the Case of $k = 0$

In the special case of zero light source elevation angle $El$, we took two images with different light source azimuth angle $Az$, for each one of the ten faces present in the database. One image with $El = -10^\circ$ and the other with $El = -25^\circ$. Our algorithm estimates depths of each one of the ten faces by solving Eq. (11). We shift and stretch the estimated depths, so that they will have the same mean and variance as the "ground truth" depths. In first part of Table 2, we provide quality estimates of the results of our special case algorithm, on all the ten faces.

We have done further alignment between the estimated and "ground truth" depths. We have normalized all the rows of the estimated depths to have the same mean as their counterpart rows in the "ground truth" depths. Thereafter we have measured the quality estimates of the results, and presented them in the second part of Table 2. One can note a significant increase in estimates, relatively to the first part of Table 2, which is an indication of a significant 1D ambiguity which is left in the solution of Eq. (11). Quality estimates in the second part of Table 2 are comparable with those of our main results in Table 1.

## 5 Conclusions

In this paper we have presented a successful combination of two previous facial shape reconstruction approaches - one which uses symmetry and one which uses

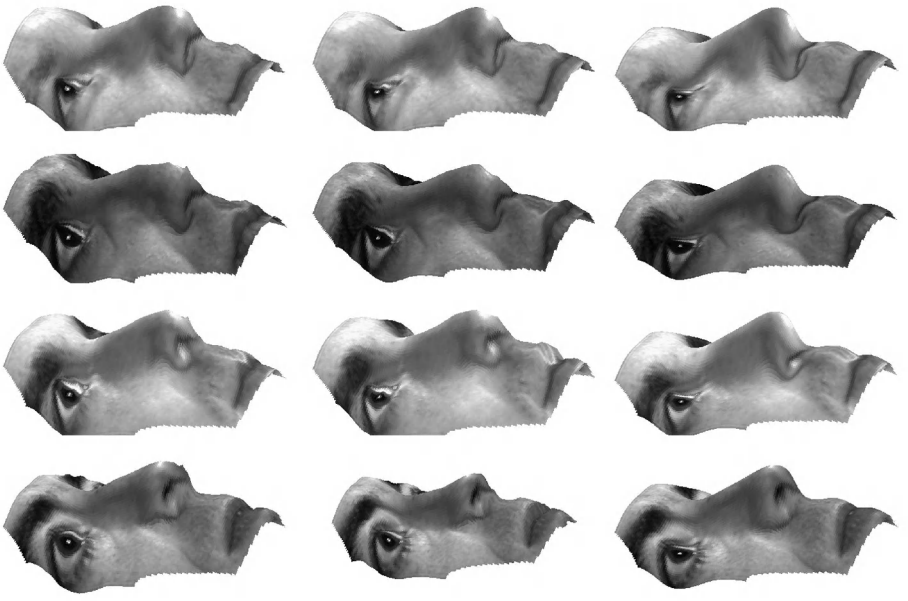


**Fig. 4.** Rows correspond to different Yale faces. First, second and third columns contain renderings of the faces with "ground truth", reconstructed and average depths, respectively. For the first three faces, texture matches rather well both "ground truth" and estimated depths, but poorly the average depth (mainly in the area between nose and mouth), indicating good shape estimation of our algorithm, at least for these faces.

statistics of human faces. Although our setup in this paper is rather restrictive and results are inaccurate on many faces, still our approach has a major advantages over the previous methods - it is very simple, provides a closed-from solution, accounts for facial nonuniform albedo and has extremely low computational complexity. The main disadvantage of the algorithm is inaccurate results on some faces caused by asymmetry of these faces. On most faces, however, we obtain reconstructions of sufficient quality for creation of realistically looking new synthetic views (see new geometry synthesis in Fig. 4 and new illumination synthesis in Fig. 5). In general, synthesizing views with new illumination does not require very accurate depth information, so that our algorithm can be considered appropriate for this application because of its simplicity and efficiency.

**Acknowledgements.** We are grateful to Prof. Sudeep Sarkar, University of South Florida, for allowing us to use the USF DARPA HumanID 3D Face Database for this research. Research was supported in part by the European Community grant number IST-2000-26001 and by the Israel Science Foundation grants number 266/02. The vision group at the Weizmann Inst. is supported in part by the Moross Laboratory for Vision Research and Robotics.

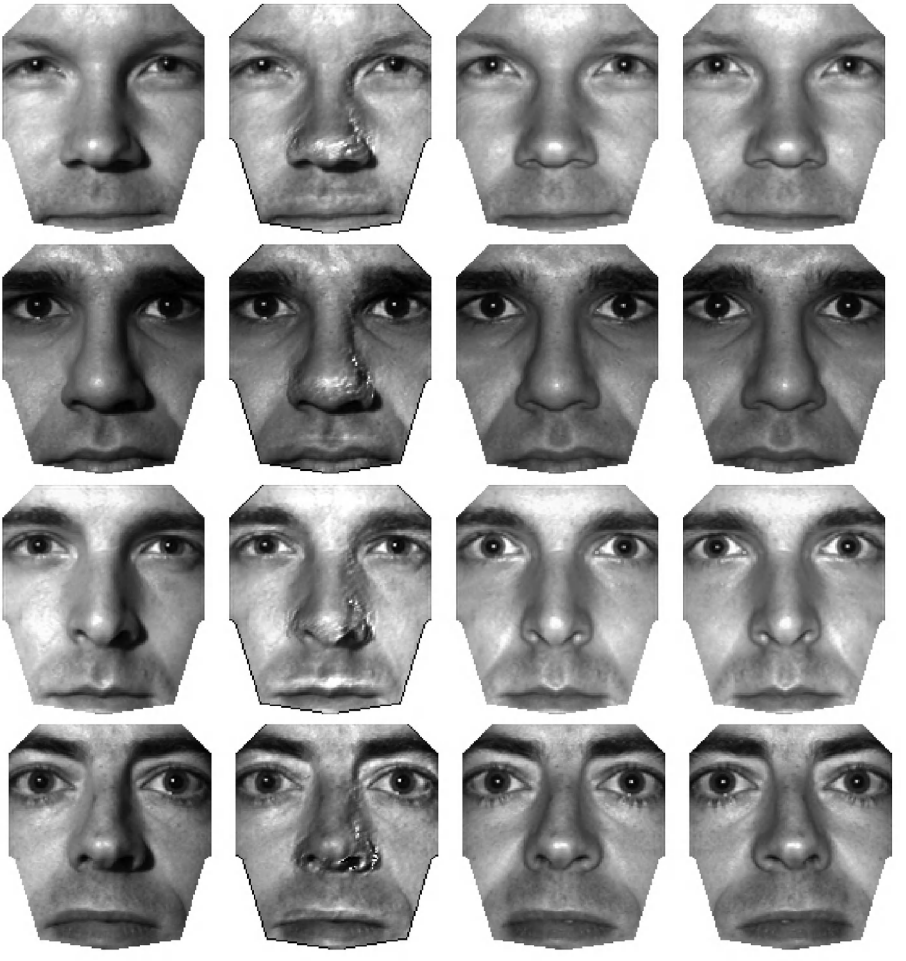


**Fig. 5.** The four different rows correspond to four different faces from the Yale face database B. The first column contains renderings of faces with side illumination $Az = 20^\circ$ and $El = 10^\circ$. The second column contains images rendered from images in column 1 using the depth recovered by our algorithm. The faces in the second column should be similar to the frontally illuminated faces in column 3 (one should ignore shadows present in the rendered images, because such a simple cancellation scheme is not supposed to cancel them out). Finally, the last column contains frontally illuminated faces from column 3, flipped around their vertical axis. By comparing two last columns, we can see noticeable facial asymmetry, even in the case of the three best faces. For the fourth face asymmetry is rather significant, specially depth asymmetry near the nose, causing rather big errors in the reconstructed depth, as can be seen in Fig. 4.

## References


1. Horn, B.K.P.: Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View. MIT AI-TR-232, 1970.
2. Zhang, R., Tsai, P.S., Cryer, J.E., Shah, M.: Shape from Shading: A Survey. IEEE Trans. on PAMI **21**(8), 690–706, 1999.
3. Shimshoni, I., Moses, Y., Lindenbaum, M.: Shape Reconstruction of 3D Bilaterally Symmetric Surfaces. IJCV **39**(2), 97–110, 2000.
4. Zhao, W., Chellappa, R.: Robust Face Recognition using Symmetric Shape-from-Shading. University of Maryland, CARTR-919, 1999.
5. Zhao, W., Chellappa, R.: Illumination-Insensitive Face Recognition using Symmetric Shape-from-Shading. In Proc. of CVPR, 286–293, 2000.
6. Lee, C.H., Rosenfeld, A.: Albedo estimation for scene segmentation. Pattern Recognition Letters **1**(3), 155–160, 1982.
7. Tsai, P.S., Shah, M.: Shape from Shading with Variable Albedo. Optical Engineering, 1212–1220, 1998.
8. Nandy, D., Ben-Arie, J.: Shape from Recognition and Learning: Recovery of 3-D Face Shapes. In Proc. of CVPR **2**, 2–7, 1999.
9. Nandy, D., Ben-Arie, J.: Shape from Recognition and Learning: Recovery of 3-D Face Shapes. IEEE Trans. On Image Processing **10**(2), 206–218, 2001.
10. Ononye, A.E., Smith, P.W.: Estimating the Shape of A Surface with Non-Constant Reflectance from a Single Color Image. In Proc. BMVC, 163–172, 2002.
11. Atick, J.J., Griffin, P.A., Redlich, A.N.: Statistical Approach to Shape from Shading: Reconstruction of Three-Dimensional Face Surfaces from Single Two-Dimensional Images. Neural Computation **8**(6), 1321–1340, 1996.
12. Blanz, V., Vetter, T.A.: A morphable model for the synthesis of 3d faces. In Proc. of SIGGRAPH, 187–194, 1999.
13. Romdhani, S., Blanz, V., Vetter, T.: Face Identification by Fitting a 3D Morphable Model using Linear Shape and Texture Error Functions. In Proc. of ECCV **4**, 3–19, 2002.
14. Blanz, V., Romdhani, S., Vetter, T.: Face Identification across different Poses and Illuminations with a 3D Morphable Model. In Proc. of the IEEE Int. Conf. on Automatic Face and Gesture Recognition, 202–207, 2002.
15. Romdhani, S., Vetter, T.: Efficient, Robust and Accurate Fitting of a 3D Morphable Model. In Proc. of ICCV, 59–66, 2003.
16. Pentland, A.P.: Finding the Illuminant Direction. JOSA **72**, 448–455, 1982.
17. Liu, Y., Schmidt, K., Cohn, J., Mitra, S.: Facial Asymmetry Quantification for Expression Invariant Human Identification. Computer Vision and Image Understanding **91**(1/2), 138–159, 2003.
18. Basri, R., Jacobs, D.W.: Photometric Stereo with General, Unknown Lighting. In Proc. of CVPR, 374-381, 2001.
19. Yilmaz, A., Shah, M.: Estimation of Arbitrary Albedo and Shape from Shading for Symmetric Objects. In Proc. of BMVC, 728-736, 2002.
20. Pentland, A.P.: Linear Shape from Shading. IJCV **4**(2), 153–162, 1990.
21. Cryer, J., Tsai, P., Shah, M.: Combining shape from shading and stereo using human vision model. University of Central Florida, CSTR-92-25, 1992.
22. Jacobs, D.W., Belhumeur, P.N., Basri, R.: Comparing images under variable illumination. In Proc. of CVPR, 610–617, 1998.
23. Zauderer, E.: Partial Differential Equations of Applied Mathematics, 2nd Edition, John Wiley and Sons, 1983.

24. Jollife, I.T.: Principal Component Analysis. Springer-Verlag, New York, 1986.
25. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience **3**(1), 71–86, 1991.
26. USF DARPA Human-ID 3D Face Database, Courtesy of Prof. Sudeep Sarkar, University of South Florida, Tampa, FL. http://marthon.csee.usf.edu/HumanID/
27. Lee, D.D., Seung, H.S.: Learning the Parts of Objects by Non-Negative Matrix Factorization. Nature **401**, 788–791, 1999.
28. Hyvaerinen, A., Oja, E.: Independent Component Analysis: Algorithms and Applications. Neural Networks **13**(4), 411–430, 2000.
29. Bartlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face recognition by independent component analysis. IEEE Trans. on Neural Networks **13**(6), 1450–1464, 2002.
30. Cheng, C.M., Lai, S.H., Chang. K.Y.: A PDM Based Approach to Recovering 3D Face Pose and Structure from Video. In Proc. of Int. Conf. on Information Technology, 238–242, 2003.
31. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From Few To Many: Generative Models For Recognition Under Variable Pose and Illumination. In Proc. of IEEE Int. Conf. on Automatic Face and Gesture Recognition, 277–284, 2000.
32. Jin, H., Favaro, P., Soatto, S.: Real-Time Feature Tracking and Outlier Rejection with Changes in Illumination. In Proc. of ICCV, 684–689, 2001.

# Region-Based Segmentation on Evolving Surfaces with Application to 3D Reconstruction of Shape and Piecewise Constant Radiance

Hailin Jin[1], Anthony J. Yezzi[2], and Stefano Soatto[1]

[1] Computer Science Department, University of California, Los Angeles, CA 90095, hljin,soatto@cs.ucla.edu
[2] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, ayezzi@ece.gatech.edu

**Abstract.** We consider the problem of estimating the shape and radiance of a scene from a calibrated set of images under the assumption that the scene is Lambertian and its radiance is piecewise constant. We model the radiance segmentation explicitly using smooth curves on the surface that bound regions of constant radiance. We pose the scene reconstruction problem in a variational framework, where the unknowns are the surface, the radiance values and the segmenting curves. We propose an iterative procedure to minimize a global cost functional that combines geometric priors on both the surface and the curves with a data fitness score. We carry out the numerical implementation in the level set framework.

**Keywords:** variational methods, Mumford-Shah functional, image segmentation, multi-view stereo, level set methods, curve evolution on manifolds.

**Fig. 1.** *(COLOR) A plush model of "nemo." The object exhibits piecewise constant appearance. From a set of calibrated views, our algorithm can estimate the shape and the piecewise constant radiance.*

## 1 Introduction

Inferring three-dimensional shape and appearance of a scene from a collection of images has been a central problem in Computer Vision, known as multi-view stereo. Traditional approaches to this problem first match points or small

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 114–125, 2004.
© Springer-Verlag Berlin Heidelberg 2004

**Fig. 2.** *Man-made objects often exhibit piecewise constant appearance. Approximating their radiances with smooth functions would lead to gross error and "blurring" of the reconstruction. On the other hand, these objects are not textured enough to establish dense correspondence among different views. However, we can clearly see radiance boundaries that divided the objects into constant regions.*

image regions across different views and then combine the matches into a three-dimensional model[1]. Scene radiances can be reconstructed afterwards if necessary. These approaches effectively avoid directly estimating the scene radiances, which can be quite complex for real scenes. However, for these methods to work, the scene has to satisfy quite restrictive assumptions, namely the radiance has to contain "sufficient texture." When the assumptions are not fulfilled, traditional methods fail. Recently, various approaches have been proposed to fill the gaps where the assumptions underlying traditional stereo methods are violated and the scene radiance is assumed to be smooth, for instance [1,2]. In this case, radiance is modeled explicitly rather than being annihilated through image-to-image matching. The problem of reconstructing shape and radiance is then formulated as a joint segmentation of all the images. The resulting algorithms are very robust to image noise.

However, there are certainly many scenes whose radiances are not heavily textured, but are not smooth either. For instance, man-made objects are often built with piecewise constant material properties and therefore exhibit approximately piecewise constant radiances, for instance those portrayed in Figure 2. For scenes like these, neither the assumption of having global constant or smooth radiances is satisfied, nor their radiances are textured enough to establish dense correspondence among different views.

For such scenes, one may attempt to use the algorithms designed for smooth radiances to reconstruct pieces of the surface that satisfy the assumptions and then patch them together to get the whole surface. Unfortunately, this approach does not work because patches are not closed surfaces, but even if they were, individual patches would not be able to explain the image data due to self-occlusions (see for instance Figure 4). Therefore, more complete and "global" models of the scene radiance are necessary. Our choice is to model it as a piecewise constant function. Under this choice, we can divide the scene into regions such that each region supports a uniform radiance, and the radiance is discontinuous across regions. The scene reconstruction problem now amounts to estimating the surface shape, the segmentation of the surface and the radiance value of each region.

[1] There are of course exceptions to this general approach, as we will discuss shortly.

### 1.1 Relation to Prior Work and Contributions

This work falls in the category of multi-view stereo reconstruction. The literature on this topic is too large to review here, so we only report on closely related work. Faugeras and Keriven [3] were the first to combine image matching and shape reconstruction in a variational framework and carry out the numerical implementation using level set methods [4]. The underlying principle of their approach is still based on image-to-image matching and therefore their algorithm works for scenes that contain significant texture. Yezzi et. al. [1] and Jin et. al. [2] approached the problem by modeling explicitly a (simplified) model of image formation, and reconstruct both shape and radiance of the scene by matching images to the underlying model, rather than to each other directly. The class of scenes they considered is Lambertian with constant or smooth radiances. In this paper, we extend their work by allowing scenes to have *discontinuous* radiances and *model explicitly the discontinuities*. In the work of shape carving by Kutulakos and Seitz [5], matching is based on the notion of photoconsistency, and the largest shape that is consistent with all the images is recovered. We use a different assumption, namely that the radiance is piecewise constant, to recover a different representation (the smoothest shape that is photometrically consistent with the data in a variational sense) as well as photometry. Since we estimate curves as radiance discontinuities, this work is related to stereo reconstruction of space curves [6,7]. The material presented in this paper is also closely related to a wealth of contributions in the field of image segmentation, particularly region-based segmentation, starting from Mumford and Shah's pioneering work [8] and including [9,10].

We use curves on surfaces to model the discontinuities of the radiance. We use level set methods [4] to evolve both the surface and the curve to perform optimization. Our curve evolution scheme is closely related to [11,12,13].

We address the problem of multi-view stereo reconstruction for Lambertian objects that have piecewise constant radiances. To the best of our knowledge we are the first to address this problem. Our solution relies *not* on matching image-to-image, but on matching all images to the underlying model of both geometry and photometry.

For scenes that satisfy the assumptions, we reconstruct (1) the shape of the scene (a collection of smooth surfaces) and the radiance of the scene, which includes (2) the segmentation of the scene radiance, defined by smooth curves, and (3) the radiance value of each region. Our implementation contains several novel aspects, including simultaneously evolving curves (radiance discontinuities) on evolving surfaces, both of which are represented by level set functions.

## 2 A Variational Formulation

We model the scene as a collection of smooth surfaces and a background. We denote collectively with $S \subset \mathbb{R}^3$ all the surface, i.e., we allow $S$ to have multiple connected components. We denote with $\mathbf{X} = [X, Y, Z]^T$ the coordinates of a generic point on $S$ with respect to a fixed reference frame. We assume to be able

to measure $n$ images of the scene $I_i : \Omega_i \to \mathbb{R}, i = 1, 2, \ldots, n$, where $\Omega_i$ is the domain of each image with area element $d\Omega_i$[2]. Each image is obtained with a calibrated camera which, after pre-processing, can be modeled as an ideal perspective projection $\pi_i : \mathbb{R}^3 \to \Omega_i; \mathbf{X} \mapsto \mathbf{x}_i \doteq \pi_i(\mathbf{X}) = \pi(\mathbf{X}_i) = [X_i/Z_i, Y_i/Z_i]^T$, where $\mathbf{X}_i = [X_i, Y_i, Z_i]^T$ are the coordinates for $\mathbf{X}$ in the $i$-th camera reference frame. $\mathbf{X}$ and $\mathbf{X}_i$ are related by a rigid body transformation, which can be represented in coordinates by a rotation matrix $R_i \in SO(3)$[3] and a translation vector $T_i \in \mathbb{R}^3$, so that $\mathbf{X}_i = R_i\mathbf{X} + T_i$. We assume that the background, denoted with $B$, covers the field of view of every camera. Without loss of generality, we assume $B$ to be a sphere with infinite radius, which can therefore be represented using angular coordinates $\Theta \in \mathbb{R}^2$. We assume that the background supports a radiance function $h : B \to \mathbb{R}$ and the surface supports another radiance function $\rho : S \to \mathbb{R}$. We define the region $Q_i \doteq \pi_i(S) \subset \Omega_i$ and denote its complement by $Q_i^c \doteq \Omega_i \backslash Q_i$.

Our assumption is that the foreground radiance $\rho$ is a *piecewise constant function*. We refer the reader to [14] for an extension to *piecewise smooth* radiances. For simplicity, the background radiance $h$ is assumed to be constant, although extensions to smooth, piecewise constant or piecewise smooth functions can be conceived. Furthermore, we assume that the discontinuities of $\rho$ can be modeled as a smooth closed curve $C$ *on the surface* $S$, and $C$ partitions $S$ into two regions $D_1$ and $D_2$ such that $D_1 \cup D_2 = S$. Note that we allow each region $D_i$ to have multiple disconnected components. Extensions to more regions are straightforward, for instance following the work of Vese and Chan [15]. We can thus re-define $\rho$ as follows:

$$\rho(\mathbf{X}) = \rho_i \in \mathbb{R} \qquad \text{for } \mathbf{X} \in D_i, i = 1, 2. \tag{1}$$

We denote with $\pi_i(D_1)$ and with $\pi_i(D_2)$ the projections of $D_1$ and $D_2$ in the $i$-th image respectively.

### 2.1 The Cost Functional

The task is to reconstruct $S$, $C$, $\rho_1$, $\rho_2$, and $h$ from the data $I_i, i = 1, 2, \ldots, n$. In order to do so, we set up a cost, $E_{data}$, that measures the discrepancy between the prediction of the unknowns and the actual measurements. Since some unknowns, namely the surface $S$ and the curve $C$, live in infinite-dimensional spaces, we need to impose regularization to make the inference problem well-posed. In particular, we assume that both the surface and the curve are smooth (geometric priors), and we leverage on our assumption that the radiance is constant within each domain. The final cost is therefore the sum of three terms:

$$E(S, C, \rho_1, \rho_2, h) = E_{data} + \alpha E_{surf} + \beta E_{curv}, \tag{2}$$

[2] More precisely, measured images are usually non-negative discrete functions defined on a discrete grid and have minimum and maximum values. For ease of notation, we will consider them to be defined on continuous domains and take values from the whole real line.

[3] $SO(3) = \{R \mid R \in \mathbb{R}^{3\times3} \text{ s.t. } R^T R = I \text{ and } \det(R) = 1\}$.

where $\alpha, \beta \in \mathbb{R}^+$ control the relative weights among the terms. The data fitness can be measured in the sense of $\mathcal{L}^2$ as:

$$E_{data} = \sum_{i=1}^{n} \left( \int_{\pi_i(D_1)} (I_i(\mathbf{x}_i) - \rho_1)^2 d\Omega_i + \int_{\pi_i(D_2)} (I_i(\mathbf{x}_i) - \rho_2)^2 d\Omega_i \right) + \sum_{i=1}^{n} \int_{Q_i^c} (I_i(\mathbf{x}_i) - h)^2 d\Omega_i, \tag{3}$$

although other function norms would do as well. The geometric prior for $S$ is given by the total surface area:

$$E_{surf} = \int_S dA, \tag{4}$$

and that for $C$ is given by the total curve length:

$$E_{curv} = \int_C ds, \tag{5}$$

where $dA$ is the Euclidean area form of $S$ and $s$ is the arc-length parameterization for $C$. Therefore, the total cost takes the expression:

$$\begin{aligned} &E_{total}(S, C, \rho_1, \rho_2, h) = E_{data} + \alpha E_{surf} + \beta E_{curv} \\ &= \sum_{i=1}^{n} \left( \int_{\pi_i(D_1)} (I_i(\mathbf{x}_i) - \rho_1)^2 d\Omega_i + \int_{\pi_i(D_2)} (I_i(\mathbf{x}_i) - \rho_2)^2 d\Omega_i \right) \\ &+ \sum_{i=1}^{n} \int_{Q_i^c} (I_i(\mathbf{x}_i) - h)^2 d\Omega_i + \alpha \int_S dA + \beta \int_C ds. \end{aligned} \tag{6}$$

This functional is in the spirit of the Mumford-Shah functional for image segmentation [8].

## 3 Optimization of the Cost Functional

In order to find the surface $S$, the radiances $\rho_1, \rho_2, h$ and the curve $C$ that minimize the cost (6), we set up an iterative procedure where we start from a generic initial condition (typically a big cube, sphere or cylinder) and update the unknowns along their gradient directions until convergence to a (necessarily local) minimum.

### 3.1 Updating the Surface

The gradient descent flow for the surface geometric prior is given by $S_t = 2\kappa N$, where $\kappa$ is the mean curvature and $N$ is the unit normal to $S$. Note that we have kept 2 in the expression in order to have the weights in the final flow match the weights in the cost (2). To facilitate computing the variation of the rest terms

with respect to the surface, we introduce the radiance characteristic function $\phi$ to describe the location of $C$ for a given surface $S$. We define $\phi : S \to \mathbb{R}$ such that

$$D_1 = \{\mathbf{X} \mid \phi(\mathbf{X}) > 0\}, \quad D_2 = \{\mathbf{X} \mid \phi(\mathbf{X}) < 0\}, \quad \text{and } C = \{\mathbf{X} \mid \phi(\mathbf{X}) = 0\}. \tag{7}$$

$\phi$ can be viewed as the level set function of $C$. However, one has to keep in mind that $\phi$ is defined on $S$. We can then express the curve length as $\int_C ds = \int_S \|\nabla_S \mathcal{H}(\phi)\| dA$ where $\mathcal{H}$ is the Heaviside function. We prove in the technical report [14] that the gradient descent flow for the curve smoothness term has the following expression

$$S_t = \frac{\delta(\phi)}{\|\nabla_S \phi\|} \mathbf{II}(\nabla_S \phi \times N) N, \tag{8}$$

where $\mathbf{II}(\mathbf{t})$ denotes the second fundamental form of a vector $\mathbf{t} \in T_P(S)$, i.e. the normal curvature along $\mathbf{t}$ for $\|\mathbf{t}\| = 1$ and $\delta$ denotes the one-dimensional Dirac distribution: $\delta = \dot{\mathcal{H}}$. $T_P(S)$ is the tangent space for $S$ at $P$. Note that $\nabla_S \phi \times N \perp N$ and therefore $\nabla_S \phi \times N \in T_P(S)$. Since flow (8) involves $\delta(\phi)$, it only acts on the places where $\phi$ is zero, i.e., the curve $C$. To find the variation of the data fitness term with respect to $S$, we need to introduce two more terms. Let $\chi_i : S \to \mathbb{R}$ be the surface visibility function with respect to the $i$-th camera, i.e. $\chi_i(\mathbf{X}) = 1$ for points on $S$ that are visible from the $i$-th camera and $\chi_i(\mathbf{X}) = 0$ otherwise. Let $\sigma_i$ be the change of coordinates from $d\Omega_i$ to $dA$, i.e, $\sigma_i \doteq \frac{d\Omega_i}{dA} = \langle \mathbf{X}_i, N_i \rangle / Z_i^3$, where $N_i$ the unit normal $N$ expressed in the $i$-th camera reference frame. We now can express the data term as follows (see the technical report [14] for more details):

$$\begin{aligned} &\sum_{i=1}^{n} \left( \int_{\pi_i(D_1)} (I_i(\mathbf{x}_i) - \rho_1)^2 d\Omega_i + \int_{\pi_i(D_2)} (I_i(\mathbf{x}_i) - \rho_2)^2 d\Omega_i + \int_{Q_i^c} (I_i(\mathbf{x}_i) - h)^2 d\Omega_i \right) \\ &= \sum_{i=1}^{n} \int_S \chi_i \Gamma_i \sigma_i dA + \sum_{i=1}^{n} \int_{\Omega_i} (I_i(\mathbf{x}_i) - h)^2 d\Omega_i, \end{aligned} \tag{9}$$

where $\Gamma_i \doteq \mathcal{H}(\phi)(I_i - \rho_1)^2 + (1 - \mathcal{H}(\phi))(I_i - \rho_2)^2 - (I_i - h)^2$. Note that we have dropped the arguments for $I_i$ and $\phi$ for ease of notation. Since, for a fixed $h$, $\sum_{i=1}^{n} \int_{\Omega_i} (I_i(\mathbf{x}_i) - h)^2 d\Omega_i$ does not depend upon the unknown surface, we only need to compute the variation of the first term $\sum_{i=1}^{n} \int_S \chi_i \Gamma_i \sigma_i dA$ with respect to $S$. We prove in the technical report [14] that the gradient descent flow for minimizing cost functionals of a general type $\sum_{i=1}^{n} \int_S \chi_i \Gamma_i \sigma_i dA$ takes the form:

$$S_t = \sum_{i=1}^{n} \frac{1}{Z_i^3} \Big( \Gamma_i \left\langle \chi_{i\mathbf{X}}, R_i^T \mathbf{X}_i \right\rangle - \chi_i \left\langle \Gamma_{i\mathbf{X}}, R_i^T \mathbf{X}_i \right\rangle \Big) N, \tag{10}$$

where $\chi_{i\mathbf{X}}$ and $\Gamma_{i\mathbf{X}}$ denote the derivatives of $\chi_i$ and $\Gamma_i$ with respect to $\mathbf{X}$ respectively. We further note that $\left\langle I_{i\mathbf{X}}, R_i^T \mathbf{X}_i \right\rangle = 0$ [16] and obtain

$$\left\langle \Gamma_{i\mathbf{X}}, R_i^T \mathbf{X}_i \right\rangle = \delta(\phi) \big( (I_i - \rho_1)^2 - (I_i - \rho_2)^2 \big) \left\langle \nabla_S \phi, R_i^T \mathbf{X}_i \right\rangle. \tag{11}$$

Therefore, the whole gradient descent flow for the cost (6) is given by

$$S_t = \Big( \sum_{i=1}^{n} \frac{\Gamma_i}{Z_i^3} \left\langle \chi_i \mathbf{X}, R_i^T \mathbf{X}_i \right\rangle - \chi_i \frac{\delta(\phi)}{Z_i^3} \big( (I_i - \rho_1)^2 - (I_i - \rho_2)^2 \big) \left\langle \nabla_S \phi, R_i^T \mathbf{X}_i \right\rangle + 2\alpha\kappa + \beta \frac{\delta(\phi)}{\|\nabla_S \phi\|} \mathbf{II}(\nabla_S \phi \times N) \Big) N. \tag{12}$$

Note that flow (12) depends only upon the image values, *not the image gradients*. This property greatly improves the robustness of the resulting algorithm to image noise when compared to other variational approaches [3] to stereo based on image-to-image matching (i.e. less prone to become "trapped" in local minima).

### 3.2 Updating the Curve

We show in the technical report [14] that the gradient descent flow for $C$ related to the smoothness of the curve is given by

$$C_t = \Big( \sum_{i=1}^{n} \big( (I_i - \rho_2)^2 - (I_i - \rho_1)^2 \big) \sigma_i + \beta \kappa_g \Big) \mathbf{n}, \tag{13}$$

where $\kappa_g$ is the geodesic curvature and $\mathbf{n}$ is the normal to the curve in $T_P(S)$ (commonly referred to as the *intrinsic normal* to the curve $C$). Since $\mathbf{n} \in T_P(S)$, $C$ stays in $S$ as it evolves according to equation (13).

### 3.3 Updating the Radiances

Finally, the optimization with respect to the radiances can be solved in closed forms as:

$$\begin{cases} \rho_1 = \dfrac{\sum_{i=1}^{n} \int_{\pi_i(D_1)} I_i(\mathbf{x}_i) d\Omega_i}{\sum_{i=1}^{n} \int_{\pi_i(D_1)} d\Omega_i} \\ \rho_2 = \dfrac{\sum_{i=1}^{n} \int_{\pi_i(D_2)} I_i(\mathbf{x}_i) d\Omega_i}{\sum_{i=1}^{n} \int_{\pi_i(D_2)} d\Omega_i} \\ h = \dfrac{\sum_{i=1}^{n} \int_{Q_i^c} I_i(\mathbf{x}_i) d\Omega_i}{\sum_{i=1}^{n} \int_{Q_i^c} d\Omega_i}, \end{cases} \tag{14}$$

i.e., the optimal values are the sample averages of the intensity values in corresponding regions.

## 4 A Few Words on the Numerical Implementation

In this section, we report some details on implementing the proposed algorithm. Both the surface and curve evolutions are carried out in the level set framework [4]. Since there has been a lot of work on shape reconstruction using level set methods and the space is limited, we refer the interested reader to [2,3,17] for general issues. We would like to point out that we do not include the term (8)

in the implementation of the surface evolution because experimental testing has empirically shown that convinging results can be obtained even neglecting this term given its localized influence only near the segmenting curve. The numerical implementation of equations (14) should be also straightforward, since one only needs to compute the sample average of the intensities in the regions $\pi_i(D_1)$, $\pi_i(D_2)$ and $Q_i^c$. Therefore, we will devote the rest of this section to issues related to the implementation of the flow (13).

Note that flow (13) is not a simple planar curve evolution. The curve is defined on the unknown surface, and therefore its motion has to be constrained on the surface. (It does not make sense to move the curve freely in $\mathbb{R}^3$, which would lead the curve out of the surface.) The way we approach the problem is to exploit the radiance characteristic function $\phi$. Our approach is similar to the one considered by [11,12]. Recall that $C$ is the zeros of $\phi$. We can express all the terms related to $C$ using $\phi$. In particular, the geodesic curvature is given by (we refer the reader to [14] for details on deriving this expression and the rest equations in this section):

$$\begin{aligned}\kappa_g &= \nabla_S \cdot \left(\frac{\nabla_S \phi}{\|\nabla_S \phi\|}\right) = \frac{\nabla_S \cdot \nabla_S \phi}{\|\nabla_S \phi\|} - \left\langle \nabla_S \phi, \nabla_S \left(\frac{1}{\|\nabla_S \phi\|}\right)\right\rangle \\ &= \frac{\Delta_S \phi}{\|\nabla_S \phi\|} - \frac{\nabla_S^T \phi \nabla_S^2 \phi \nabla_S \phi}{\|\nabla_S \phi\|^3},\end{aligned} \tag{15}$$

where $\nabla_S^2 \phi$ and $\Delta_S \phi$ denote the intrinsic Hessian and the intrinsic Laplacian of $\phi$ respectively. After representing the curve $C$ with $\phi$, we can implement the curve evolution by evolving the function $\phi$ on the surface. We further relax[4] $\phi$ from being a function defined on $S$ to being a function defined on $\mathbb{R}^3$. This is related to the work of [13] for smoothing functions on surfaces. We denote with $\varphi$ the extended function. We can then express the intrinsic gradient as follows:

$$\nabla_S \phi = \nabla \varphi - \langle \nabla \varphi, N \rangle N, \tag{16}$$

and the intrinsic Hessian as follows:

$$\nabla_S^2 \phi = (I - NN^T)\nabla^2 \varphi (I - NN^T) - (N^T \nabla \varphi) \frac{(I - NN^T)\nabla^2 \psi (I - NN^T)}{\|\nabla \psi\|}, \tag{17}$$

where $\nabla^2$ stands for the standard Hessian in space and $\psi$ is the level set function for $S$. $\Delta_S \phi$ can be computed as

$$\Delta_S \phi = \operatorname{trace}(\nabla_S^2 \phi) = \Delta \varphi - 2\kappa N^T \nabla \varphi - N^T \nabla^2 \varphi N. \tag{18}$$

Finally the curve evolution (13) is given by updating the following partial differential equation

$$\phi_t = \|\nabla_S \phi\| \sum_{i=1}^{n} \big((I_i - \rho_2)^2 - (I_i - \rho_1)^2\big)\sigma_i + \beta\Big(\Delta_S \phi - \frac{\nabla_S^T \phi \nabla_S^2 \phi \nabla_S \phi}{\|\nabla_S \phi\|^2}\Big), \tag{19}$$

[4] This relaxation does not necessarily have to cover the entire $\mathbb{R}^3$. It only needs to cover the regions where the numerical computation operates.

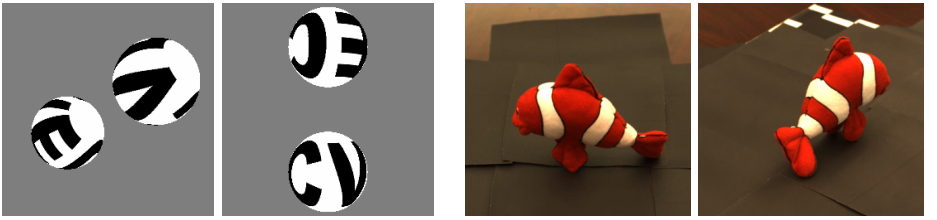


**Fig. 3.** *(COLOR) The left* 2 *images are* 2 *out of* 26 *views of a synthetic scene. The scene consists of two spheres, each of which is painted in black with the word "ECCV". The rest of the spheres is white and the background is gray. Each image is of size* $257 \times 257$*. The right* 2 *images are* 2 *out of* 31 *images from the "nemo" dataset. Each image is of size* $335 \times 315$ *and calibrated manually using a calibration rig.*

with $\phi$ replaced by $\varphi$ and $\nabla_S \phi$, $\nabla_S^2 \phi$ and $\Delta_S \phi$ replaced by the corresponding terms of $\varphi$ according to equations (16), (17) and (18).

## 5 Experiments

In Figure 3 (left 2 images) we show 2 out of 26 views of a synthetic scene, which consists of two spheres. Each image is of size $257 \times 257$. Each sphere is painted in black with the word "ECCV" and the rest is white. The background is gray. Clearly modeling this scene with one single constant radiance would lead to gross errors. One cannot even reconstruct either white or black part using the smooth radiance model in [1,2] due to occlusions. For comparison purpose, we report the results of our implementation of [1] in Figure 4 (the right 2 images). The left 4 images in Figure 4 show the final reconstructed shape using the proposed algorithm. The red curve is where the discontinuities of the radiance are. The explicit modeling of radiance discontinuities may enable further applications. For instance, one can flatten the surface and the curve and perform character recognition of the letters. The numerical grid used in both algorithms is the same and of size $128 \times 128 \times 128$. In Figure 5 we show the surface evolving from a large cylinder to a final solid model. The foreground in all the images is rendered with its estimated radiance values ($\rho_1$ and $\rho_2$) and the segmenting curve is rendered in red. In Figure 6 we show the images reconstructed using the estimated surface, radiances and segmenting curve compared with an actual image from the original dataset.

In Figure 3 (right 2 images) we show 2 out of 31 views of a real scene, which contains a plush model of "nemo". The intrinsics and extrinsics of all the images are calibrated off-line. Each image is of size $335 \times 315$. Nemo is red with white stripes. For the proposed algorithm to work with color images, we have extended the model (6) as follows: we consider images to take vector values (RGB color in our case) and modify the square error between scalars in equation (6) to the simple square of Euclidean vector norm. In Figure 7 we show several shaded views of the final reconstructed shape using the proposed algorithm. The

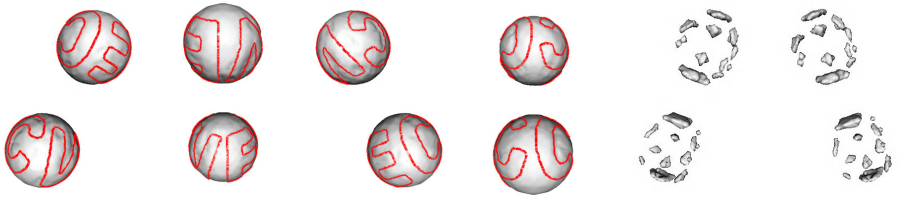


**Fig. 4.** *(COLOR) The first* 4 *images are shaded views of the final shape estimated using the proposed algorithm. Radiance discontinuities have been rendered as red curves. The location of the radiance discontinuities can be exploited for further purposes, for instance character recognition. The last* 2 *images are the results of assuming that the foreground has constant radiance, as in [1]. Note that the algorithm of [1] cannot capture all the white parts or all the black parts of the spheres, because that is not consistent with the input images due to occlusions.*

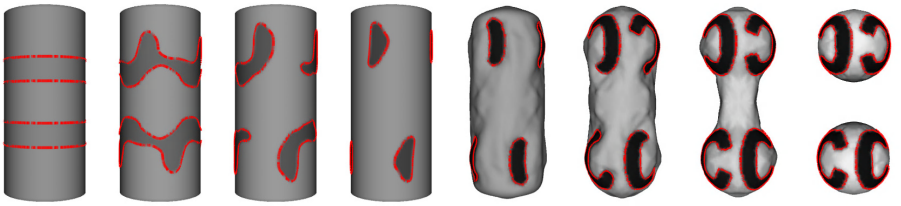


**Fig. 5.** *(COLOR) Rendered surface during evolution. The foreground in all the images is rendered with the current estimate of the radiance ($\rho_1$ and $\rho_2$) plus some shading effects for ease of visualization.*

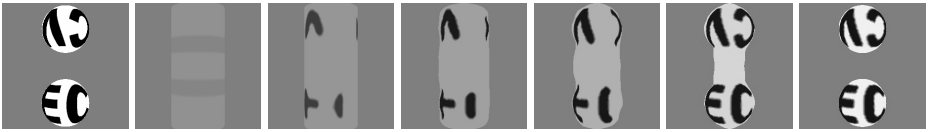


**Fig. 6.** *The first image is just one view from the original dataset. The remaining* 6 *images are rendered using estimates from different stages of the estimation process. In particular, the second image is rendered using the initial data and the last image is rendered using the final estimates.*

radiance discontinuities are rendered as green curves. The numerical grid used here is of size $128 \times 60 \times 100$. In Figure 8 we show the surface evolving from an initial shape that neither contains nor is contained in the shape of the scene, to a final solid model. The foreground in all the images is rendered with its estimated radiance values ($\rho_1$ and $\rho_2$) and the segmenting curve is rendered in green. In Figure 9 we show the images reconstructed using the estimated surface, radiances and segmenting curve compared with one actual image in the original dataset.

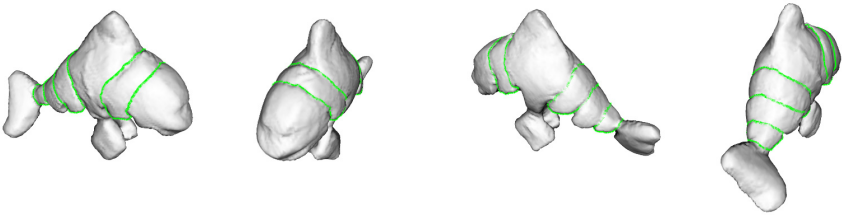


**Fig. 7.** *(COLOR) Several shaded views of the final reconstructed surface. The radiance discontinuities have been highlighted in green.*

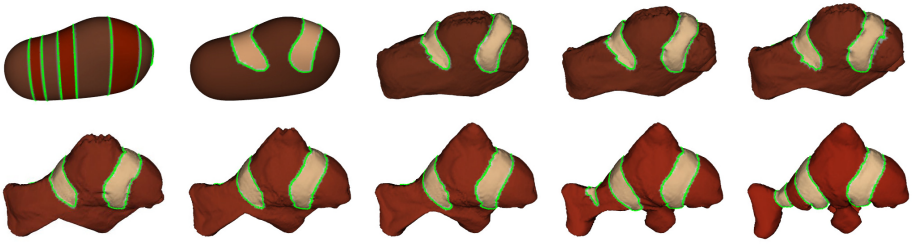


**Fig. 8.** *(COLOR) Rendered surface during evolution. Notice that the initial surface is neither contained nor contains the actual object. The foreground in all the images are rendered with the current estimate of the radiance values ($\rho_1$ and $\rho_2$) plus some shading effects for ease of visualization.*

**Fig. 9.** *(COLOR) The first image is just one view from the original dataset. The remaining* 6 *images are rendered using estimates from different stages of the estimation process. In particular, the second image is rendered using the initial data and the last image is rendered using the final estimates.*

## 6 Conclusions

We have presented an algorithm to reconstruct the shape and radiance of a Lambertian scene with piecewise constant radiance from a collection of calibrated views. We set the problem in a variational framework and minimize a cost functional with respect to the unknown shape, unknown radiance values in each region, and unknown radiance discontinuities. We use gradient-descent partial differential equations to simultaneously evolve a surface in space (shape), a curve defined on the surface (radiance discontinuities) and radiance values of each region, which are implemented numerically using level set methods.

**Acknowledgements.** This work is supported by AFOSR grant F49620-03-1-0095, NSF grants IIS-0208197, CCR-0121778 and CCR-0133736, and ONR grant N00014-03-1-0850. We would like to thank Daniel Cremers for stimulating discussions and Li-Tien Cheng for suggestions on implementing curve evolution on surfaces.

## References

1. Yezzi, A.J., Soatto, S.: Stereoscopic segmentation. In: Proc. of Intl. Conf. on Computer Vision. Volume 1. (2001) 59–66
2. Jin, H., Yezzi, A.J., Tsai, Y.H., Cheng, L.T., Soatto, S.: Estimation of 3d surface shape and smooth radiance from 2d images: A level set approach. J. Scientific Computing **19** (2003) 267–292
3. Faugeras, O., Keriven, R.: Variational principles, surface evolution, pdes, level set methods, and the stereo problem. IEEE Trans. on Image Processing **7** (1998) 336–344
4. Osher, S.J., Sethian, J.A.: Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. J. Comput. Phys. **79** (1988) 12–49
5. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. Int. J. of Computer Vision **38** (2000) 199–218
6. Kahl, F., August, J.: Multiview reconstruction of space curves. In: Proc. of Intl. Conf. on Computer Vision. Volume 2. (2003) 1017–1024
7. Schmid, C., Zisserman, A.: The geometry and matching of lines and curves over multiple views. Int. J. of Computer Vision **40** (2000) 199–233
8. Mumford, D., Shah, J.: Optimal approximations by piecewise smooth functions and associated variational problems. Comm. Pure Appl. Math. **42** (1989) 577–685
9. Chan, T.F., Vese, L.A.: Active contours without edges. IEEE Trans. on Image Processing **10** (2001) 266–277
10. Yezzi, A.J., Tsai, A., Willsky, A.: A statistical approach to snakes for bimodal and trimodal imagery. In: Proc. of Intl. Conf. on Computer Vision. Volume 2. (1999) 898–903
11. Cheng, L.T., Burchard, P., Merriman, B., Osher, S.J.: Motion of curves constrained on surfaces using a level-set approach. J. Comput. Phys. **175** (2002) 604–644
12. Kimmel, R.: Intrinsic scale space for images on surfaces: the geodesic curvature flow. Graphical Models and Image Processing **59** (1997) 365–372
13. Bertalmio, M., Cheng, L., Osher, S.J., Sapiro, G.: Variational problems and partial differential equations on implicit surfaces. J. Comput. Phys. **174** (2001) 759–780
14. Jin, H., Yezzi, A.J., Soatto, S.: Region-based segmentation on evolving surfaces with application to 3d reconstruction of shape and piecewise smooth radiance. Technical Report CSD-TR04-0004, University of California at Los Angeles (2004)
15. Vese, L.A., Chan, T.F.: A multiphase level set framework for image segmentation using the mumford and shah model. Int. J. of Computer Vision **50** (2002) 271–293
16. Soatto, S., Yezzi, A.J., Jin, H.: Tales of shape and radiance in multi-view stereo. In: Proc. of Intl. Conf. on Computer Vision. Volume 1. (2003) 171–178
17. Jin, H., Soatto, S., Yezzi, A.J.: Multi-view stereo beyond lambert. In: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition. Volume 1. (2003) 171–178

# Human Upper Body Pose Estimation in Static Images

Mun Wai Lee and Isaac Cohen

Institute for Robotics and Intelligent Systems
Integrated Media Systems Center
University of Southern California
Los Angeles, CA 90089-0273, USA
{munlee, icohen}@usc.edu
http://www-scf.usc.edu/~munlee/index.html

**Abstract.** Estimating human pose in static images is challenging due to the high dimensional state space, presence of image clutter and ambiguities of image observations. We present an MCMC framework for estimating 3D human upper body pose. A generative model, comprising of the human articulated structure, shape and clothing models, is used to formulate likelihood measures for evaluating solution candidates. We adopt a data-driven proposal mechanism for searching the solution space efficiently. We introduce the use of proposal maps, which is an efficient way of implementing inference proposals derived from multiple types of image cues. Qualitative and quantitative results show that the technique is effective in estimating 3D body pose over a variety of images.

## 1 Estimating Pose in Static Image

This paper proposes a technique for estimating human upper body pose in static images. Specifically, we want to estimate the 3D body configuration defined by a set of parameters that represent the global orientation of the body and body joint angles. We are focusing on middle resolution images, where a person's upper body length is about 100 pixels or more. Images of people in meetings or other indoor environment are usually of this resolution. We are currently only concerned with estimating the upper body pose, which is relevant for indoor scene. In this situation the lower body is often occluded and the upper body conveys most of a person's gestures. We do not make any restrictive assumptions about the background and the human shape and clothing, except for not wearing any head wear nor gloves.

### 1.1 Issues

There are two main issues in pose estimation with static images, the high dimension state space and pose ambiguity.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 126–138, 2004.
© Springer-Verlag Berlin Heidelberg 2004

**High Dimension State Space.** Human upper body pose has about 20 parameters and pose estimation involves searching in a high dimensional space with complex distribution. With static images, there is no preceding pose for initializing the search, unlike a video tracking problem. This calls for an efficient mechanism for exploring the solution space. In particular, the search is preferably data-driven, so that good solution candidates can be found easily.

**Pose Ambiguity.** From a single view, the inherent non-observability of some of the degrees of freedom in the body model leads to *forwards/backwards flipping ambiguities* [10] of the depth positions of body joints. Ambiguity is also caused by noisy and false observations. This problem can be partly alleviated by using multiple image cues to achieve robustness.

### 1.2 Related Work

Pose estimation on video has been addressed in many previous works, either using multiple cameras [3] or a single camera [2,9]. Many of these works used the particle filter approach to estimate the body pose over time, by relying on a good initialization and temporal smoothness. Observation-based importance sampling scheme has also been integrated into this approach to improve robustness and efficiency [5].

For static images, some works have been reported for recognizing prototype body poses using shape context descriptors and exemplars [6]. Another related work involves the mapping of image features into body configurations [8]. These works however rely on either a clean background or that the human is segmented by a background subtraction and therefore not suitable for fully automatic pose estimation in static images.

Various reported efforts were dedicated to the detection and localization of body parts in images. In [4,7], the authors modeled the appearance and the 2D geometric configuration of body parts. These methods focus on real-time detection of people and do not estimate the 3D body pose. Recovering 3D pose was studied in [1,11], but the proposed methods assume that image positions of body joints are known and therefore tremendously simplify the problem.

## 2 Proposed Approach

We propose to address this problem, by building an image generative model and using the MCMC framework to search the solution space. The image generative model consists of (*i*) human model, which encompasses the articulated structure, shape and the type of clothing, (*ii*) scene-to-image projection, and (*iii*) generation of image features. The objective is to find the human pose that maximizes the posterior probability.

We use the MCMC technique to sample the complex solution space. The set of solution samples generated by the Markov chain weakly converges to a stationary distribution equivalent to the posterior distribution. Data-driven MCMC

framework [13] allows us to design good proposal functions, derived from image observations. These observations include face, head-shoulder contour, and skin color blobs. These observations, weighted according to their saliency, are used to generate *proposal maps*, that represent the proposal distributions of the image positions of body joints. These maps are first used to infer solutions on a set of 2D pose variables, and subsequently generate proposals on the 3D pose using inverse kinematics. The proposal maps considerably improve the estimation, by consolidating the evidences provided by different image cues.

## 3 Human Model

### 3.1 Pose Model

This model represents the articulated structure of the human body and the degree of freedom in human kinematics. The upper body consists of 7 joints, 10 body parts and 21 degree of freedom (6 for global orientation and 15 for joint angles). We assume an orthographic projection and use a scale parameter to represent the person height.

### 3.2 Probabilistic Shape Model

The shape of each body part is approximated by a truncated 3D cone. Each cone has three free parameters: the length of the cone and the widths of the top and base of the cone. The aspect ratio of the cross section is assumed to be constant. Some of the cones share common widths at the connecting joints. In total, there are 16 shape parameters. As some of the parameters have small variances and some are highly correlated, the shape space is reduced to 6 dimensions using PCA, and this accounts for 95% of the shape variation in the training data set.

### 3.3 Clothing Model

This model describes the person's clothing to allow the hypothesis on where the skin is visible, so that observed skin color features can be interpreted correctly. As we are only concerned with the upper body, we use a simple model with only one parameter that describes the length of the sleeve. For efficiency, we quantized this parameter into five discrete levels, as shown in Figure 1a.

## 4 Prior Model

We denote the state variable as $m$, which consists of four subsets: ($i$) global orientation parameters: $g$, ($ii$) local joint angles: $j$, ($iii$) human shape parameters: $s$, and ($iv$) clothing parameter: $c$.

$$m = \{g, i, s, c\} \ . \tag{1}$$

Assuming that the subsets of parameters are independent, the prior distribution of the state variable is given by:

$$p(m) \approx p(g)p(j)p(s)p(c). \tag{2}$$

**Global Orientation Parameters.** The global orientation parameters consist of image position ($x_g$), rotation parameters ($r_g$) and a scale parameter ($h_g$). We assume these parameters to be independent so that the following property holds:

$$p(g) \approx p(x_g)p(r_g)p(h_g). \tag{3}$$

The prior distributions are modeled as normal distributions and learned from training data.

**Joint Angles Parameters.** The subset $j$, consists of 15 parameters describing the joint angles at 7 different body joint locations.

$$j = \{j_i, i = neck, left_wrist, left_elbow, \ldots, right_shoulder\}. \tag{4}$$

In general, the joint angles are not independent. However, it is impracticable to learn the joint distribution of the 15 dimensional $j$ vector, with a limited training data set. As an approximation, our prior model consists of joint distribution of pair-wise neighboring body joint locations. For each body location, $i$, we specify a neighboring body location as its parent, where:

$$\begin{array}{ll} parent(left_wrist) = left_elbow & parent(right_wrist) = right_elbow \\ parent(left_elbow) = left_shoulder & parent(right_elbow) = right_shoulder \\ parent(left_shoulder) = torso & parent(right_shoulder) = torso \\ parent(neck) = torso & parent(torso) = \emptyset \end{array}$$

The prior distribution is then approximated as:

$$p(j) \approx \lambda_{pose} \prod_i p(j_i) + (1 - \lambda_{pose}) \prod_i p(j_i, j_{parent(i)}) \tag{5}$$

where $\lambda_{pose}$ is a constant valued between 0 and 1. The prior distributions $p(j_i)$ and $p(j_i, j_{parent(i)})$ are modeled as Gaussians. The constant $\lambda_{pose}$ is estimated from training data using cross-validation, based on the maximum likelihood principle.

**Shape Parameters.** PCA is used to reduce the dimensionality of the shape space by transforming the variable $s$ to a 6 dimensions variable $s'$ and the prior distribution is approximated by a Gaussian:

$$p(s) \approx p(s') \approx N(s', \mu_{s'}, \Sigma_{s'}) \tag{6}$$

where $\mu_{s'}$ and $\Sigma_{s'}$ are the mean and covariance matrix of the prior distribution of $s'$.

**Clothing Parameters.** The clothing model consists of a discrete variable $c$, representing the sleeve length. The prior distribution is based on the empirical frequency in the training data.

**Marginalized distribution of image positions of body joints.** We denote $\{u_i\}$ as the set of image positions of body joints. Given a set of parameters $\{g, j, s\}$, we are able to compute the image position of each body joints $\{u_i\}$:

$$u_i = f_i(g, j, s) \tag{7}$$

where $f_i(.)$ is a deterministic forward kinematic function. Therefore, there exists a prior distribution for each image position:

$$p(u_i) = \int\int\int f_i(g, j, s)p(g)p(j)p(s)dgdjds \tag{8}$$

where $p(u_i)$ represents the marginalized prior distribution of the image position of the $i$-th body joint. In fact, any variable that is derived from image positions of the body joints has a prior distribution, such as the lengths of the arms in the image or the joint positions of the hand and elbow. As will be described later, these prior distributions are useful in computing weights for the image observations. The prior distribution of these measures can be computed from Equation (8) or it can be learned directly from the training data as was performed in our implementation.

## 5 Image Observations

Image observations are used to compute data driven proposal distribution in the MCMC framework. The extraction of observations consists of 3 stages: (*i*) face detection, (*ii*) head-shoulders contour matching, and (*iii*) skin blobs detection.

### 5.1 Face Detection

For face detection, we use the Adaboost technique proposed by [12]. We denote the face detection output as a set of face candidates,

$$I_{Face} = \{I_{Face_Position}, I_{Face_Size}\}, \tag{9}$$

where $I_{Face_Position}$ is the detected face location and $I_{Face_Size}$ is the estimated face size. The observation can be used to provide a proposal distribution for the image head position, $u_{Head}$, modeled as a Gaussian distribution:

$$q(u_{Head}|I_{Face}) \sim N(u_{Head} - I_{Face_Position}, \cdot, \cdot). \tag{10}$$

The parameters of the Gaussian are estimated from training data. The above expression can be extended to handle multiple detected faces.

### 5.2 Head-Shoulder Contour Matching

**Contour Model for Head-Shoulder.** We are interested in detecting 2D contour of the head and shoulders. Each contour is represented by a set of connected

points. This contour is pose and person dependent. For robustness, we use a mixture model approach to represent the distribution of the 2D contour space. Using a set of 100 training data, a K-mean clustering algorithm is used to learn the means of 8 components, as shown in Figure 1b. The joint distributions of these contour and the image position of head, neck and shoulders are also learned from the training data.

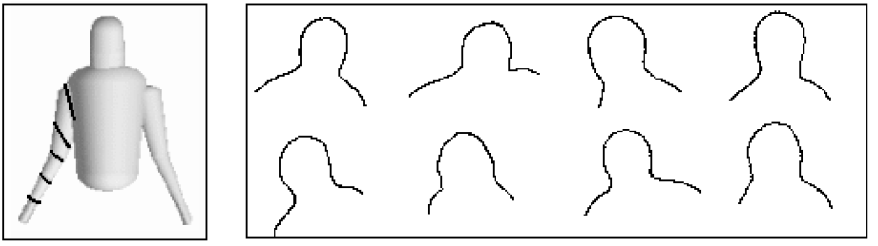



**Fig. 1.** Models: (a) Quantized sleeve length of clothing model, (b) components of head-shoulder model.

**Contour Matching.** In each test image, we extract edges using the Canny detector, and a gradient descent approach is used to align each exemplar contour to these edges. We define a search window around a detected face, and initiate searches at different positions within this window. This typically results in about 200 contour candidates. The confidence of each candidate is weighted based on (*i*) confidence weight of the detected face, (*ii*) joint probability of contour position and detected face position, and (*iii*) edge alignment error. The number of candidates is reduced to about 50, by removing those with low confidence.

The resulting output is a set of matched contours $\{ I_{Head_Shoulder,i}\}$. Each contour provides observations on the image positions of the *head*, *neck*, *left shoulder* and *right shoulder*, with a confidence weight $w_{HS,i}$:

$$I_{Head_Shoulder,i} = \{w_{HS,i}, I_{Head_Pos,i}, I_{Neck_Pos,i}, I_{L_Shoulder_Pos,i}, I_{R_Shoulder_Pos,i}\}. \quad (11)$$

Each observation is used to provide proposal candidates for the image positions of the head ($u_{Head}$), left shoulder ($u_{L_Shoulder}$), right shoulder ($u_{R_Shoulder}$), and neck ($u_{Neck}$). The proposal distributions are modeled as Gaussian distributions given by:

$$\begin{aligned} &q(u_{Head}|I_{Head_Shoulder,i}) \sim w_{HS,i}N(u_{Head} - I_{Head_Pos,i}, \cdot, \cdot) \\ &q(u_{Neck}|I_{Head_Shoulder,i}) \sim w_{HS,i}N(u_{Neck} - I_{Neck_Pos,i}, \cdot, \cdot) \\ &q(u_{L_Shoulder}|I_{Head_Shoulder,i}) \sim w_{HS,i}N(u_{L_Shoulder} - I_{L_Shoulder_Pos,i}, \cdot, \cdot) \\ &q(u_{R_Shoulder}|I_{Head_Shoulder,i}) \sim w_{HS,i}N(u_{R_Shoulder} - I_{R_Shoulder_Pos,i}, \cdot, \cdot) \end{aligned} \quad (12)$$

The approach to combine all these observations is described in Section 5.4.

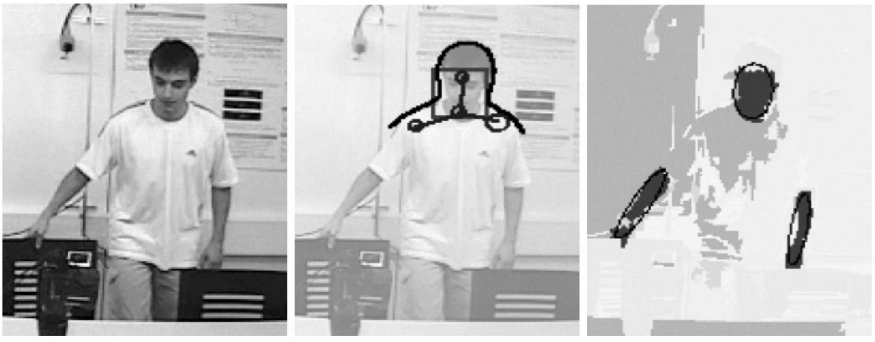


**Fig. 2.** Image observations, from left: (i) original image, (ii) detected face and head-shoulders contour, (iii) skin color ellipses extraction.

### 5.3 Elliptical Skin Blob Detection

Skin color features provide important cues on arms positions. Skin blobs are detected in four sub-stages: (*i*) color based image segmentation is applied to divide the image into smaller regions, (*ii*) the probability of skin for each segmented region is computed using a histogram-based skin color Bayesian classifier, (*iii*) ellipses are fitted to the boundaries of these regions to form skin ellipse candidates, and (*iv*) adjacent regions with high skin probabilities are merged to form larger regions (see Figure 2).

The extracted skin ellipses are used for inferring the positions of limbs. The interpretation of a skin ellipse is however dependent on the clothing type. For example, if the person is wearing short sleeves, then the skin ellipse represent the lower arm, indicating the hand and elbow positions. However, for long sleeve, the skin ellipse should cover only the hand and used for inferring the hand position only. Therefore the extracted skin ellipses provide different sets of interpretation depending on the hypothesis on the clothing type in the current Markov chain state.

For clarity in the following description, we assume that the clothing type is short sleeve. For each skin ellipse, we extract the two extreme points of the ellipse along the major axis. These points are considered as plausible candidates for the hand-elbow pair, or elbow-hand pair of either the left or right arm. Each candidate is weighted by (*i*) skin color probability of the ellipse, (*ii*) likelihood of the arm length, (*iii*) joint probability of the elbow, hand positions with one of the shoulder candidates (For each ellipse, we find the best shoulder candidate that provides the highest joint probability.)

### 5.4 Proposal Maps

In this section we present the new concept of proposal maps. Proposal maps are generated from image observation to represent the proposal distributions of the image positions of body joints. For this discussion, we focus on the generation of a proposal map for the left hand. Using the skin ellipse cues presented earlier, we generate a set of hypotheses on the left hand position, $\{I_{L_Hand,i}, i = 1, \ldots, N_h\}$, where $N_h$ is the number of hypotheses. Each hypothesis has an associated weight $w_{L_Hand,i}$ and a covariance matrix $\Sigma_{L_Hand,i}$ representing the measurement uncertainty. From each hypothesis, the proposal distribution for the left hand image position is given by:

$$q(u_{L_Hand}|I_{L_Hand,i}) \propto w_{L_Hand,i} N(u_{L_Hand}, I_{L_Hand,i}, \Sigma_{L_Hand,i}). \quad (13)$$

Contributions of all the hypotheses are combined as follows:

$$q(u_{L_Hand}|\{I_{L_Hand,i}\}) \propto \max_i q(u_{L_Hand}|I_{L_Hand,i}). \quad (14)$$

As the hypotheses are, in general, not independent, we use the *max* function instead of the *summation* in Equation (14); otherwise peaks in the proposal distribution would be overly exaggerated. This proposal distribution is unchanged throughout the MCMC process. To improve efficiency, we approximate the distribution as a discrete space with samples corresponding to every pixel position. This same approach is used to combine multiple observations for other body joints. Figure 3 shows the pseudo-color representation of the proposal maps for various body joints. Notice that the proposal maps have multiple modes, especially for the arms, due to ambiguous observations and image clutter.

## 6 Image Likelihood Measure

The image likelihood $P(I|m)$ consists of two components: (*i*) a region likelihood, and (*ii*) a color likelihood. We have opted for an adaptation of the image likelihood measure introduced in [14].

**Region Likelihood.** Color segmentation is performed to divide an input image into a number of regions. Given a state variable $m$, we can compute the corresponding *human blob* in the image. Ideally, the human blob should match to the union of a certain subset of the segmented regions.

Denoting $\{R_i, i = 1, \ldots, N_{region}\}$ as the set of segmented regions, $N_{region}$ is the number of segmented regions and $H_m$ the human blob predicted from the state variable $m$. For the correct pose, each region $R_i$ should either belong to the human blob $H_m$ or to the *background* blob $\bar{H}_m$. In each segmented region $R_i$, we count the number of pixels that belong to $H_m$ and $\bar{H}_m$.

$$\begin{aligned} N_{i,human} &= \text{count pixels } (u,v) \text{ where } (u,v) \in R_i \text{ and } (u,v) \in H_m, \\ N_{i,background} &= \text{count pixels } (u,v) \text{ where } (u,v) \in R_i \text{ and } (u,v) \in \bar{H}_m. \end{aligned} \quad (15)$$

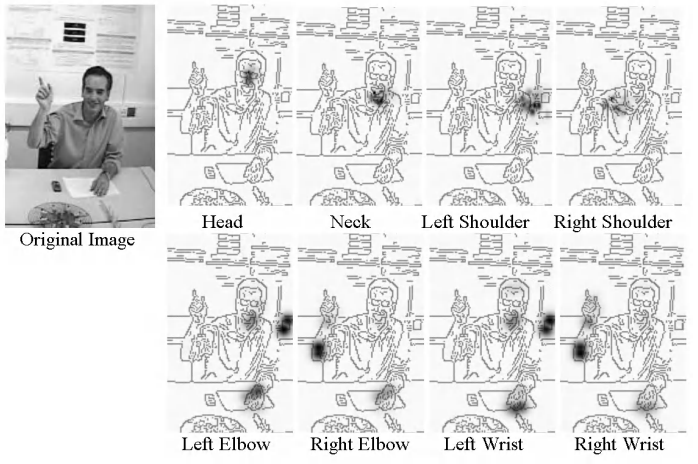



**Fig. 3.** Proposal maps for various body joints. The proposal probability of each pixel is illustrated in pseudo-color (or grey level in monochrome version).

We define a binary label, $l_i$ for each region and classify the region, so that

$$l_i = \begin{cases} 1 \text{ if } N_{i,human} \geq N_{i,background} \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{16}$$

We then count the number of *incoherent* pixels, $N_{incoherent}$, given as:

$$N_{incoherent} = \sum_{i=1}^{N_{region}} (N_{i,background})^{l_i} (N_{i,human})^{1-l_i}. \tag{17}$$

The region-based likelihood measurement is then defined by:

$$L_{region} = \exp(-\lambda_{region} N_{incoherent}) \tag{18}$$

where $\lambda_{region}$ is a constant determined empirically using a Poisson model.

**Color Likelihood.** The likelihood measure expresses the difference between the color distributions of the human blob $H_m$ and the background blob $\bar{H}_m$. Given the predicted blobs $H_m$ and $\bar{H}_m$ , we compute the corresponding color distributions, denoted by $d$ and $b$. The color distributions are expressed by normalized histograms with $N_{histogram}$ bins. The color likelihood is then defined by:

$$L_{color} = \exp(-\lambda_{color} B_{d,b}^2) \tag{19}$$

where $\lambda_{color}$ is a constant and $B_{d,b}$ is the Bhattachayya coefficient measuring the similarity of two color distributions and defined by:

$$B_{d,b} = \sum_{i=1}^{N_{histogram}} \sqrt{d_i b_i}. \tag{20}$$

The combined likelihood measure is given by :

$$P(I|m) = L_{region} \times L_{color}. \tag{21}$$

## 7 MCMC and Proposal Distribution

We adapted the data-driven MCMC framework [13]. which allows the use of image observations for designing proposal distribution to find region of high density efficiently. At the $t$-th iteration in the Markov chain process, a candidate $m'$ is sampled from $q(m_t|m_{t-1})$ and accepted with probability,

$$p = \min\left\{1, \frac{p(m'|I)q(m_{t-1}|m')}{p(m_{t-1}|I)q(m'|m_{t-1})}\right\}. \tag{22}$$

The proposal process is executed by three types of Markov chain dynamics described in the following.

**Diffusion Dynamic.** This process serves as a local optimizer and the proposal distribution is given by:

$$q(m'|m_{t-1}) \propto N(m', m_{t-1}, \Sigma_{diffusion}) \tag{23}$$

where the variance $\Sigma_{diffusion}$ is set to reflect the local variance of the posterior distribution, estimated from training data.

**Proposal Jump Dynamic.** This jump dynamic allows exploratory search across different regions of the solution space using proposal maps derived from observation. In each jump, only a subset of the proposal maps is used. For this discussion, we focus on observations of the left hand. To perform a jump, we sample a candidate of the hand position from the proposal map:

$$\hat{u}_{L_hand} \sim q(u_{L_hand}|\{I_{L_hand,i}\}). \tag{24}$$

The sampled hand image position is then used to compute, via inverse kinematics (IK), a new state variable $m'$ that satisfies the following condition:

$$f_i(m') = \begin{cases} f_i(m_{t-1}) & \text{where } j \neq L_hand \\ \hat{u}_{L_hand} & \text{where } j = L_hand \end{cases} \tag{25}$$

where $f_i(m_{t-1})$ is the deterministic function that generates image position of a body joint, given the state variable. In other words, IK is performed by keeping other joint positions constant and modify the pose parameters to adjust the image position of the left hand. When there are multiple solutions due to depth ambiguity, we choose the solution that has the minimum change in depth. If $m'$ cannot be computed (e.g. violate the geometric constraints), then the proposed candidate is rejected.

**Flip Dynamic.** This dynamic involves flipping a body part (i.e. head, hand, lower arm or entire arm) along depth direction, around its pivotal joint [10]. Flip dynamic is balanced so that forward and backward flips have the same proposal probability. The solution candidate $m'$ is computed by inverse kinematics.

## 8 Experimental Results

We used images of indoor meeting scenes as well as outdoors images for testing. Ground truth is generated by manually locating the positions of various body joints on the images and estimating the relative depths of these joints. This data set is available at http://www-scf.usc.edu/~munlee/PoseEstimation.html.

### 8.1 Pose Estimation

Figure 4 shows the obtained results on various images. These images were not among the training data. The estimated human model and its pose (solutions with the highest posterior probability) are projected onto the original image and a 3D rendering from a sideward view is also shown.

The estimated joint positions were compared with the ground truth data, and a RMS error was computed. Since the depth had higher uncertainties, we computed two separate measurements, one for the 2D positions, and the other for the depth. The histograms of these errors (18 images processed) are shown in Figure 5a. This set of images and the pose estimation results are available at the webpage: http://www-scf.usc.edu/~munlee/images/upperPoseResult.htm.

### 8.2 Convergence Analysis

Figure 5b shows the RMS errors (averaged over test images) with respect to the MCMC iterations. As the figure shows, the error for the 2D image position decreases rapidly from the start of the MCMC process and this is largely due to the observation-driven proposal dynamics. For the depth estimate, the kinematics flip dynamic was helpful in finding hypotheses with good depth estimates. It however required a longer time for exploration. The convergence time varies considerably among different images, depending on the quality of the image observations. For example, if there were many false observations, the convergence required a longer time. On average, 1000 iterations took about 5 minutes.

## 9 Conclusion

We have presented an MCMC framework for estimating 3D human upper body pose in static images. This hypothesis-and-test framework uses a generative model with domain knowledge such as the human articulated structure and allows us to formulate appropriate prior distributions and likelihood functions, for evaluating samples in the solution space.

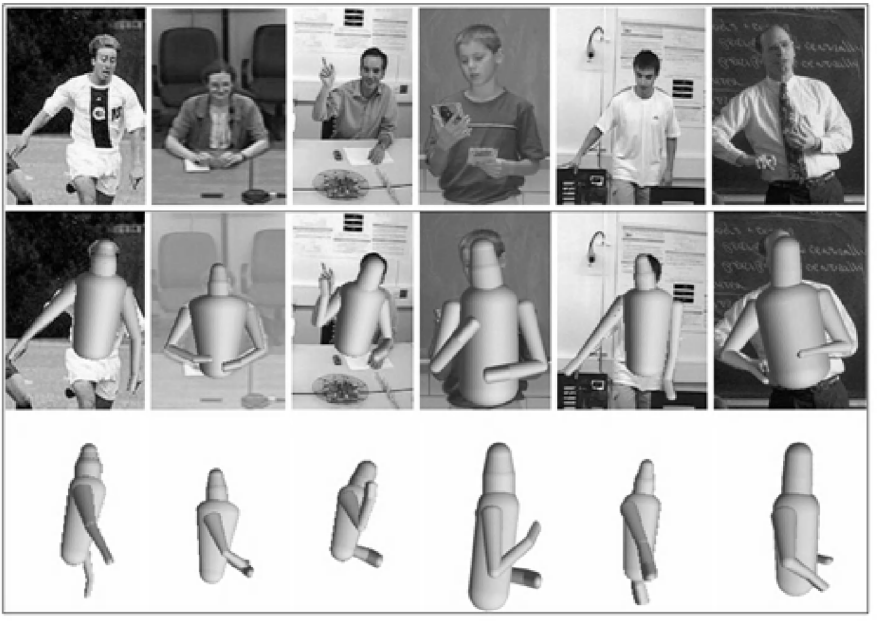


**Fig. 4.** Pose Estimation. First Row: Original images, second row: estimated poses, third row: estimated poses (side view).

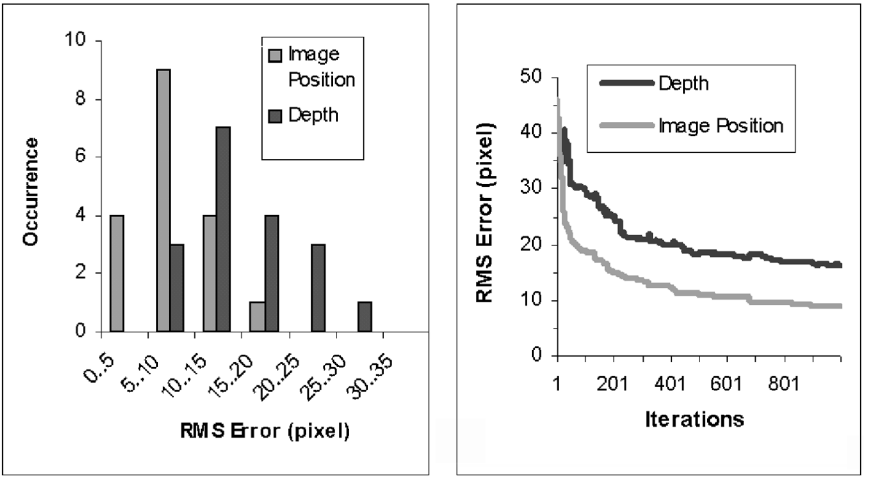



**Fig. 5.** Results: (a) Histogram of RMS Error (b) Convergence Analysis.

In addition, the concern with high dimensionality and efficiency postulates that the searching process should be more driven by image observations. The data-driven MCMC framework offers the flexibility in designing proposal mechanism for sampling the solution space. Our technique incorporates multiple cues to provide robustness.

We introduce the use of proposal map, which is an efficient way of consolidating information provided by observations and implementing proposal distributions. Qualitative and quantitative results are presented to show that the technique is effective over a wide variety of images. In future work, we will extend our work to full body pose estimation and video-based tracking.

**Acknowledgment.** This research was partially funded by the Integrated Media Systems Center, a National Science Foundation Engineering Research Center, under Cooperative Agreement No. EEC-9529152. We would like to thank the PETS workshop committee for providing images on meeting scene.

## References

1. Barron, C., Kakadiaris, I. A.: Estimating anthropometry and pose from a single image. CVPR 2000, vol.1, pp. 669-676.
2. Bregler, C., Malik, J.: Tracking people with twists and exponential maps. CVPR 1998, pp. 8-15.
3. Deutscher, J., Davison, A., Reid, I.: Automatic partitioning of high dimensional search spaces associated with articulated body motion capture. CVPR 2001, vol. 2, pp. 669-676.
4. Ioffe, S., Forsyth, D.A.: Probabilistic methods for finding people. IJCV 43(1), pp.45-68, June 2001.
5. Lee, M. W., Cohen, I.: Human Body Tracking with Auxiliary Measurements. AMFG 2003, pp.112-119.
6. Mori, G., Malik, J.: Estimating Human Body Configurations using Shape Context Matching. ECCV 2002, pp 666-680.
7. Ronfard, R., Schmid, C., Triggs, B.: Learning to parse pictures of people. ECCV 2002, vol. 4, pp. 700-714.
8. Rosales, R., Sclaroff, S.: Inferring body pose without tracking body parts. CVPR 2000, vol.2, pp. 721-727.
9. Sminchisescu, C., Triggs, B.: Covariance Scaled Sampling for Monocular 3D Body Tracking. CVPR 2001, vol.1, pp. 447-454.
10. Sminchisescu, C., Triggs, B.: Kinematic Jump Processes for Monocular Human Tracking. CVPR 2003, vol.1, pp. 69-76.
11. Taylor, C.J.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. CVIU 80(3): 349-363, December 2000.
12. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. CVPR 2001, vol.1 , pp.511-518.
13. Zhu, S., Zhang, R., Tu, Z.: Integrating bottom-up/top-down for object recognition by data driven Markov chain Monte Carlo. CVPR 2000, vol.1, pp.738-745.
14. Zhao, T., Nevatia, R.: Bayesian Human Segmentation in Crowded Situations. CVPR 2003, vol.2 pp.459-466.

# Automated Optic Disc Localization and Contour Detection Using Ellipse Fitting and Wavelet Transform

P.M.D.S. Pallawala[1], Wynne Hsu[1], Mong Li Lee[1], and Kah-Guan Au Eong[2,3]

[1] School of Computing, National University of Singapore, Singapore
{pererapa, whsu, leeml}@comp.nus.edu.sg,
[2] Ophthalmology and Visual Sciences, Alexandra Hospital, Singapore
[3] The Eye Institute, National Healthcare Group, Singapore
kah_guan_au_eong@alexhosp.com.sg

**Abstract.** Optic disc detection is important in the computer-aided analysis of retinal images. It is crucial for the precise identification of the macula to enable successful grading of macular pathology such as diabetic maculopathy. However, the extreme variation of intensity features within the optic disc and intensity variations close to the optic disc boundary presents a major obstacle in automated optic disc detection. The presence of blood vessels, crescents and peripapillary chorioretinal atrophy seen in myopic patients also increase the complexity of detection. Existing techniques have not addressed these difficult cases, and are neither adaptable nor sufficiently sensitive and specific for real-life application. This work presents a novel algorithm to detect the optic disc based on wavelet processing and ellipse fitting. We first employ Daubechies wavelet transform to approximate the optic disc region. Next, an abstract representation of the optic disc is obtained using an intensity-based template. This yields robust results in cases where the optic disc intensity is highly non-homogenous. Ellipse fitting algorithm is then utilized to detect the optic disc contour from this abstract representation. Additional wavelet processing is performed on the more complex cases to improve the contour detection rate. Experiments on 279 consecutive retinal images of diabetic patients indicate that this approach is able to achieve an accuracy of 94% for optic disc detection.

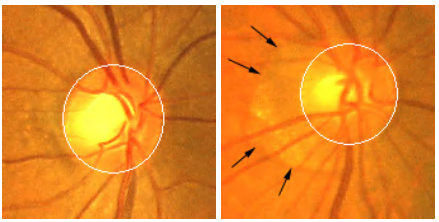
## 1 Introduction

Digital retinal images are widely used in the diagnosis and follow-up management of patients with eye disorders such as glaucoma, diabetic retinopathy, and age-related macular degeneration. Glaucoma is the second leading cause of blindness in the world, affecting some 67 to 105 million patients [20]. In glaucoma, an abnormally raised intraocular pressure damages the optic nerve and results in morphological changes in the optic disc. This leads to an increase in the size of the optic cup. Diabetic retinopathy is also a leading cause of blindness and visual impairment in many developed countries and accounts for 12,000 to 24,000 blind cases in United States alone every year [5].

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 139–151, 2004.
© Springer-Verlag Berlin Heidelberg 2004

The automated detection of optic disc has several potential clinical uses. First, the vertical diameters of the optic cup and disc may aid the diagnosis of glaucoma [3]. Changes in these parameters of the optic disc in serial images may indicate progression of the disease. Second, it allows the identification of the macula using the spatial relationship between the optic disc and macula. The macula is located on the temporal aspect of optic disc and is situated at a distance of about 2.5 disc diameters from the centre of the optic disc [9]. Occurrence of lesions in the macula region as a result of diabetic retinopathy and age-related macular degeneration are often sight-threatening. Identifying the macula allows highly sensitive algorithms to be designed to detect signs of abnormality in the macular region.

The optic disc appears as an elliptical region with high intensity in retinal images (see Fig. 1). The vertical and horizontal diameters of an optic disc are typically $1.82 \pm .15$mm and $1.74 \pm .21$ mm respectively [3]. Clinically, optic disc measurements can be obtained by approximating the disc to an ellipse [2].



(a) (b)

**Fig. 1.** (a) Outline of optic disc (white ellipse), (b) Outline of optic disc with peripapillary chorioretinal atrophy (black arrows)

While existing algorithms [8,10,13,14,15,16,18] employ a variety of techniques to detect optic disc, they are neither sufficiently sensitive nor specific enough for clinical application. The main obstacle is the extreme variation of the optic disc intensity features and the presence of retinal blood vessels (Fig. 1(a)). Peripapillary chorioretinal atrophy which are commonly seen in myopic eyes also increase the complexity of optic disc detection. This presents as a bright crescent-shaped area adjacent to the optic disc, usually on its temporal side (Fig. 1(b)), or as a bright annular (doughnut-shaped) area surrounding the optic disc.

Our proposed approach overcomes the above challenges as follows. We first approximate the optic disc boundary via the use of Daubechies wavelet transform and intensity-based techniques. Next, an ellipse fitting algorithm is employed to detect the optic disc contour in the optic disc boundary region. Experiments on 279 consecutive retinal images disclosed that we were able to achieve an accuracy of 94% for optic disc detection and 93% accuracy based on mean vertical diameter assessment.

## 2 Related Work

There has been a long stream of research to automate optic disc detection. Techniques such as active contour models [10,16], template matching [8], pyramidal decomposition [8], variance image calculation [18] and clustering techniques [13] have been developed. Among them, active contour-based models have been shown to give better results compared to the other techniques. We evaluate active contour models on optic discs ranging from least contour variants to complex variations, and discuss their result and limitations here.

Snakes or active contours [7,11,17] are curves defined within an image domain that can move under the influence of internal forces coming from the curve itself and external forces computed from the image data. There are two types of active contour models: parametric active contours [22] and geometric active contours [23]. Parametric active contours synthesize parametric curves within image domains and allow them to move towards desired features, usually edges. A traditional snake is a curve X(s) = [x(s), y(s)], s ∈ [0, 1], that moves through the spatial domain of an image to minimize the energy functional

$$E = \int_0^1 \frac{1}{2}[\alpha|\dot{x}(s)|^2 + \beta|\ddot{x}(s)|^2] + \gamma E_{ext}(x(s))ds \quad (1)$$

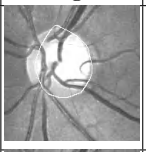
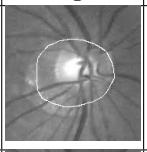
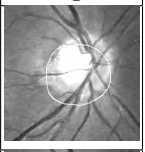
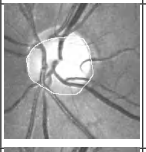
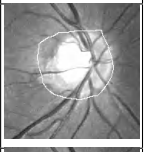
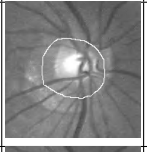
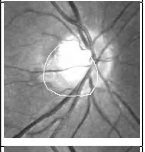
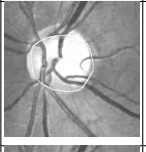
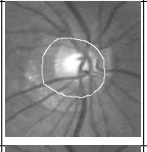
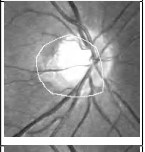
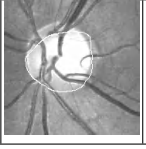
where $\alpha$, $\beta$, $\gamma$ are weighting parameters that control the snake's tension, rigidity and influence of external force, respectively, and $\dot{x}(s)$ and $\ddot{x}(s)$ denote the first and second derivatives of x(s) with respect to s. The external energy function $E_{ext}$ is derived from the image so that it takes on its smaller values at the features of interest, such as boundaries.

Analysis of the digital retinal images reveals that the use of the gradient image to derive the external energy function needed by the active contours model is not suitable because the gradient image contains too much noise arising from the retinal vessels. Even after removing retinal vessels [6] from the gradient image, it may not be complete and the removal process may introduce operator that will distort the original optic disc contour. Another option is to use intensity-based external force $E_{ext}$ model. Here, we use the gray value of the green layer of the original image as the external force. Table 1 shows the results for various weight parameters. Note that the intensity-based external force model tends to produce poorer results. Further attempts to improve results using morphological operators have not been successful due to wide variations in optic disc features.

D.T. Morris et al. [16] reported the use of active contour models to detect the optic disc with a preprocessing step to overcome these problems. Images are first preprocessed using histogram equalization. This is followed by the use of pyramid edge detector. While this approach shows improved results, it suffers from two drawbacks. First, the preprocessing steps may cause the optic disc boundary to become intractable because it fuses with the surrounding high intensity regions. Second, the pyramid edge detector is unable to filter noise from vessel edges adequately and active contour model will fail to outline the optic disc boundary correctly.

Similarly, while region snakes works well for optic discs with uniform intensity distribution, it tends to fail in optic discs having very low intensity or in cases where segment of optic disc has very low intensity. Application of deformable super-quadrics, or dynamic models with global and local deformation properties inherited from super-quadric ellipsoids and membrane splines, may be useful in optic disc detection. However, it will fail in cases with peripapillary atrophy, where there is a high intensity region next to the optic disc. Further, its high computational cost is not suitable for online processing of digital retinal images. These limitations motivated us to develop a robust yet efficient technique to reliably locate and outline the optic disc.

**Table 1.** Results for active contour models

| $\alpha$ | $\beta$ | $\gamma$ | Image 1 | Image 2 | Image 3 |
|---|---|---|---|---|---|
| 0.75 | 1.65 | 0.75 |  |  |  |
| 0.95 | 0.6 | 0.5 |  | |  |
| 0.95 | 1.4 | .75 | |  |  |
| 0.95 | 1.3 | 0.7 |  |  |  |
| 1.3 | 0.7 | 0.6 |  | | |

## 3 Optic Disc Localization and Contour Detection

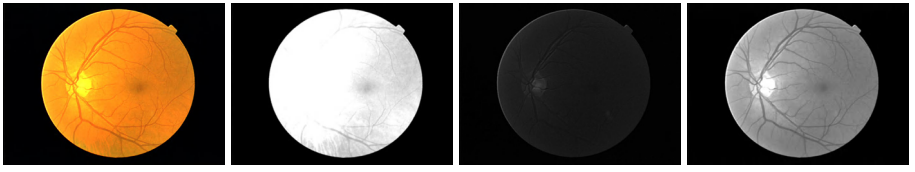
The major steps in the proposed approach to reliably detect the optic disc in large numbers of retinal images under diverse conditions are as follows. First, the approximate location of the optic disc is estimated via wavelet transform.

The intensity template is employed to construct an abstract representation of the optic disc. This abstract representation of the optic disc significantly reduces the processing area, thus increasing the computational efficiency. Next, an ellipse fitting procedure is applied to detect disc contour and to filter out difficult cases. Finally, a wavelet-based high pass filter is used to remove undesirable edge noise and to enhance the detection of non-homogenous optic discs.

Our image database consists of digital retinal images captured using a Topcon© fundus camera. All the images are standard 40-degree field of the retina centered on the macula. Image resolution is 25micron/pixel. Images were stored in 24-bit TIFF format with image size of 768*576 pixels.

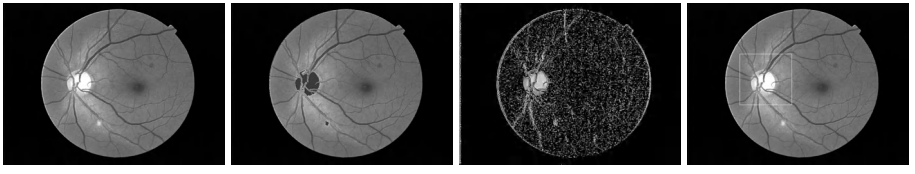
### 3.1 Localization of Optic Disc Window by Daubechies Wavelet Transformation

Figure 2 shows the different color layers of a typical retinal image. It is evident that the optic disc outline is not present in the red layer (Fig. 2(b)) or the blue layer (Fig. 2(c)). In contrast, the green layer (Fig. 2(d)) captures the optic disc outline. We use this layer for subsequent processing.



(a) Original Image (b) Red Layer (c) Blue Layer (d) Green Layer

**Fig. 2.** Color layers of a retinal image



(a) (b) (c) (d)

**Fig. 3.** Selected optic disc region using Daubechies wavelet transform

There has been a growing interest to use wavelets as a new transform technique for image processing. The aim of wavelet transform is to 'express' an input signal as a series of coefficients of specified energy. We use the Daubechies wavelet [12] to localize the optic disc. First, a wavelet transform is carried out to obtain the wavelet coefficients. Next, an inverse wavelet transform is performed after thresholding the HH component (high pass in vertical and horizontal direction) (Fig. 3(b)). The resultant image is then subtracted from the original

retinal image to obtain the subtracted image, and its sub-images (16x16 pixels) are analyzed (Fig. 3(c)).

Note that the sub-image with the highest mean value correlates to the area inside the optic disc. Hence, the center of the sub-image ($Xc,\ Yc$) with the highest mean intensity is selected and the optic disc region is defined as a $W$x$W$ window centered at ($Xc,\ Yc$). The dimension $W$ is determined by taking into consideration the image resolution (25micron/pixel) and the average size of the optic disc in the general population. Based on the results, $W$ is set to be 180. Fig. 3(d) shows the selected optic disc window. Experiments on 279 digital retinal images show 100% accuracy in the detection of optic disc.

### 3.2 Abstract Representation of Optic Disc Boundary Region

We have shown that there exist wide variations in optic disc boundary, from clear boundary outlines to very difficult cases with complex boundary outlines. To minimize the interference from these complications, we use an abstract representation of the optic to capture the optic disc boundary. This has shown to give robust results including cases with highly non-homogenous optic discs.

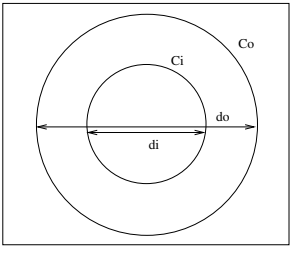



**Fig. 4.** Template to localize optic disc boundary

The abstract representation of the optic disc is in the form of a template is shown in Fig. 4. It consists of two circles: an inner circle $C_i$ and an outer circle $C_o$. $C_i$ denotes the approximated optic disc boundary and the region between the $C_i$ and $C_o$ is the immediate background. Both $C_o$ and $C_i$ are concentric circles, and the diameter ($d_o$) of $C_o$ is defined as

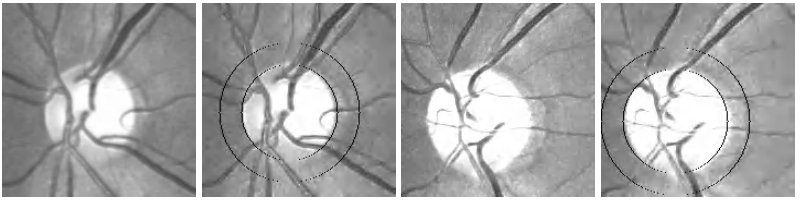
$$d_o = d_i + K \tag{2}$$

The optimal $K$ value is obtained by using a training image set. The optic disc is approximated to the template by calculating the intensity ratio ($IR$ ) as follows:

$$IR = M_i/M_o \tag{3}$$

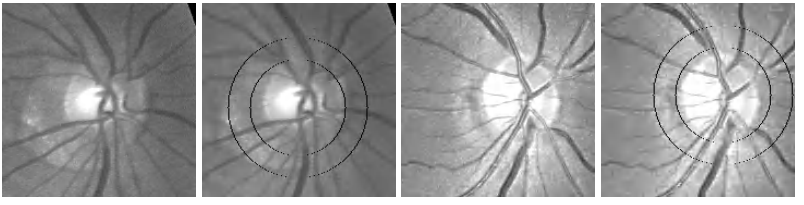
where $M_i$ is the mean intensity of pixels inside the circle $C_i$ and $M_o$ is the mean intensity of the region between circles $C_i$ and $C_o$. Vessel pixels are not involved in the calculation of mean intensity to increase the accuracy. The abstract representation of the optic disc is obtained by searching for the best fitting inner circle $C_i$. Fig. 5 and Fig. 6 show the abstract representations obtained.

The optic disc boundary region is selected as the region between $d_i \pm K$ (Fig. 7). By processing the optic disc at an abstract level rather than pixel level, we are able to detect the optic disc boundary region accurately in cases where the optic disc is highly non-homogenous.



(a) (b) (c) (d)

**Fig. 5.** (a), (c) Uniform optic disc images; (b), (d) Fitting of template



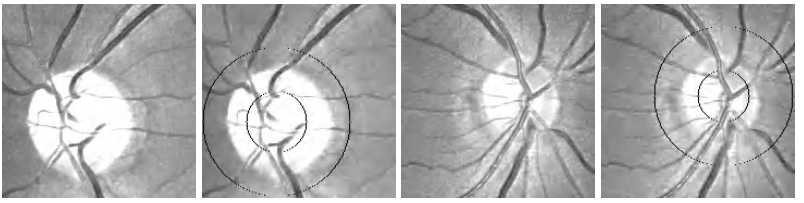
(a) (b) (c) (d)

**Fig. 6.** (a), (c) Non-uniform optic disc images; (b), (d) Fitting of template

### 3.3 Ellipse Fitting to Detect Optic Disc Contour

One of the basic tasks in pattern recognition and computer vision is the fitting of geometric primitives to a set of points. Existing ellipse fitting algorithms exploits methods such as Hough transforms [1], Kalman filtering, fuzzy clustering, or least square approach [4]. These can be divided into (1) clustering and (2) optimization based methods.

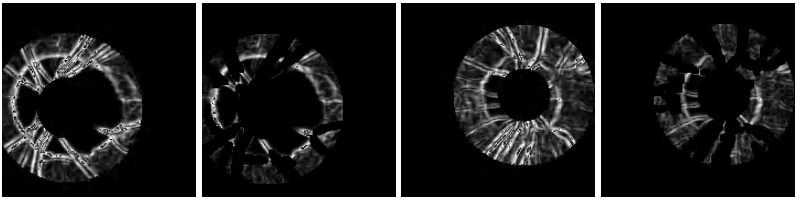
The first group of fitting techniques includes Hough transform and fuzzy clustering, which are robust against outliers and can detect multiple primitives simultaneously. Unfortunately, these techniques have low accuracy, are slow and require large amount of memory. The second group of fitting methods, which includes the Least Square approach [4], is based on the optimization of an objective function that characterizes the goodness of a particular ellipse with respect to the given set of data points. The main advantages of this group of methods



(a) (b) (c) (d)

**Fig. 7.** (a), (c) Optic disc regions; (b), (d) Isolated optic disc boundary region

are their speed and accuracy. However, these methods can fit only one primitive at a time, that is, the data should be pre-segmented before the fitting. Further, they are more sensitive to the effect of outlier compared to clustering methods.



(a) (b) (c) (d)

**Fig. 8.** (a), (c) Sobel edge maps; (b), (d) After vessel removal

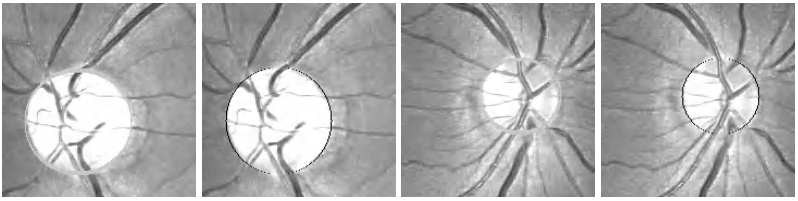
In our proposed ellipse fitting algorithm, a Sobel edge map of the optic disc boundary region is used (Fig. 8(a) and (c)). These Sobel images tend to have a high degree of noise arising from blood vessel edges and break at a number of places. Hence, we first remove all the vessel information by using a retina vessel detection algorithm [6] (Fig. 8(b) and (d)). Ellipse fitting algorithm is then used to detect optic disc contour from the resultant images.

Our ellipse fitting algorithm finds the four best fitting ellipses with minimal errors. The ellipse center is moved within the area defined by inner circle $C_i$. The ellipse major axis $a$ varies between $W/2 \pm W/4$, while the minor axis $b$ of the ellipse is restricted to $(1 \pm 0.2)*a$ pixels. These conditions are set according to the optic disc variations. The best fitting ellipses are given by

$$EF_i = P_i * (a + b) \tag{4}$$

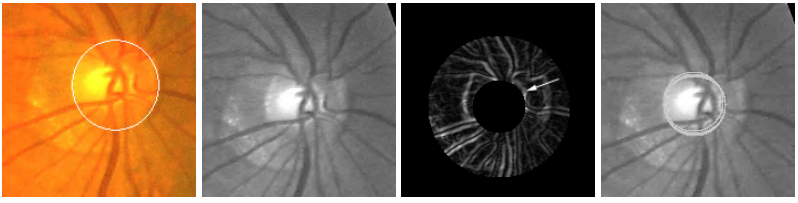
where $EF_i$ is the measure of ellipse fitting and $P_i$ is the number of edge points for the ellipse $i$. Ellipses having four highest $EF_i$ are selected and the intensity ratios for the four ellipses are calculated (see equation 3). The ellipse with the highest $IR$ whose major and minor axis falls between $(1 \pm 0.25)$ $d_i$ is regarded as the detected optic disc contour. Fig. 9 shows that the ellipse fitting procedure is able to accurately detect the optic disc boundary.

Fig. 10 depicts a difficult case where the ellipse fitting model detects part of the optic cup edge as the optic disc contour. Careful analysis reveals that this is due to the presence of optic cup edge points which tends to over-shadow the actual edge points of the optic disc boundary. In these situations, a wavelet-based enhancement is initiated.



(a) (b) (c) (d)

**Fig. 9.** (a), (c) Four best ellipses superimposed on optic disc region; (b), (d) Correctly detected ellipse

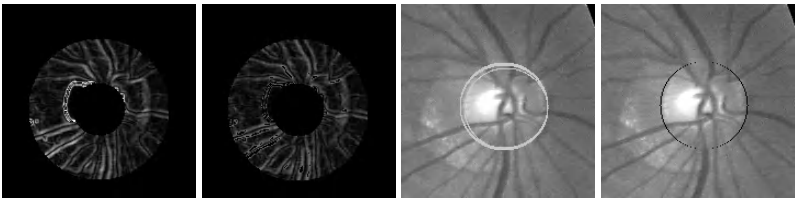


(a) (b) (c) (d)

**Fig. 10.** (a) Manual outline of optic disc; (b) Optic disc region green layer; (c) Arrow indicate optic cup edge interference; (d) Detected ellipses

### 3.4 Enhancement Using Daubechies Wavelet Transformation

To overcome the problem of noise due to the presence of optic cup points, we employ Daubechies wavelet transform [12] to enhance the optic disc boundary. This is achieved by performing the inverse wavelet transformation of coefficients after filtering out the HH component. This step gives rise to an image whose optic cup region has been removed. Fig. 11(a) shows the edge map of an inverse thresholded image. Once the edge image has been obtained, we further threshold the edge image with the image mean. This successfully removes the very prominent edge points due to optic cup and gives prominence to the faint optic disc boundary edges (Fig. 11(b)). Fig. 11(d) shows an accurately detected optic disc boundary after wavelet processing.



(a) (b) (c) (d)

**Fig. 11.** (a) Sobel edge image after wavelet enhancement; (b) Thresholding with image mean; (c) Ellipses selected by algorithm; (d) Best fitting ellipse

## 4 Experimental Results

We evaluated our proposed approach on 279 consecutive digital retinal images. The following performance criteria are used:

(1) *Accuracy* – Ratio of the number of acceptably detected contours as assessed by a trained medical doctor over the total number of images.

(2) *Vertical Diameter Assessment* – Average ratio of the vertical diameter of the detected contour over the vertical diameter of the actual optic disc boundary.

For criteria (2), the optic disc boundary outline of the images has been carefully traced by a trained medical doctor and the entire optic disc area is transformed to gray value of 255 with the background set to 0. The actual vertical diameter of the disc boundary is obtained from this transformed image.

Table 2 shows the results obtained. Without additional wavelet processing, the optic disc detection algorithm achieved 86% accuracy and 87% vertical diameter assessment. Using Daubechies wavelet processing to improve the difficult cases, we are able to achieve an accuracy of 94% and vertical diameter assessment of 93%. This improvement of 8% in accuracy includes the most difficult cases where the optic disc is of low intensity and is situated in a neighborhood with high intensity variations.

**Table 2.** Detection of optic disc contour

| | Accuracy | Vertical Diameter Assessment |
|---|---|---|
| Ellipse fitting without wavelet processing | 86% (240/279) | 87% |
| Ellipse fitting with wavelet processing | 94% (262/279) | 93% |

## 5 Discussion

Existing optic disc detection algorithms focus mainly on optic disc localization and detection of the optic disc boundary. Optic disc localization is important as it reduces the computational cost. [8] propose an optic disc localization algorithm using pyramidal decomposition. Potential optic disc regions are located using Haar wavelet-based pyramidal decomposition and are analyzed using Hausdorff template matching to detect probable optic disc. [18] design a localization algorithm based on variance of image intensity. The variance of intensity of adjacent pixels is used for recognition of the optic disc. The original retinal image is subdivided into sub-images and their respective mean intensities are calculated. Variance image is formed by a transformation which include mean of the sub-image. The location of the maximum of this image is taken as the centre of the optic disc.

[13] employs clustering techniques with simple thresholding to select several probable optic disc regions. These regions are clustered into groups and further analyzed by principle component analysis to identify the optic disc. This algorithm has yielded robust results in images with large high intensity lesions such as hard exudates in diabetic retinopathy. The drawbacks are that they are time-consuming and the results are not easily reproducible [24].

Optic disc contour detection has been attempted with active contour models [10,16] and template matching [8]. Active contour models have failed to detect optic disc contour accurately due to the presence of noise, various lesions, intensity changes close to retinal vessels, and other factors. Various preprocessing techniques have been employed to overcome these problems, including morphological filtering, pyramid edge detection, etc., but no large scale testing has been carried out to validate their accuracies. In this work, we have compared the proposed algorithm with active contour models to validate its robustness. Template matching [8] yields better results because it tends to view the optic disc as a whole entity rather than processing at pixel level. However, none of the algorithms has been tested on a large number of images and proven to be sufficiently robust and accurate for clinical use.

## 6 Conclusion

In this paper, we have presented an optic disc detection algorithm that employs ellipse fitting and wavelet processing to detect optic disc contour accurately. Experimental results have shown that the algorithm is capable of achieving 94% accuracy for the optic disc detection and 93% accuracy for the assessment of vertical optic disc diameter in 279 consecutive digital retinal images obtained from patients in a diabetic retinopathy screening program. The assessment of vertical optic disc diameter, when combined with parameters such as the vertical optic cup diameter, can provide useful information for the diagnosis and follow up management of glaucoma patients.

## References


1. Aguado, A.S., Nixon, M.S.: A New Hough Transform Mapping for Ellipse Detection. Technical Report, University of Southampton (1995)
2. Balo, K.P., Mihluedo, H., Djagnikpo, P.A., Akpandja, M.S., Bechetoille, A.: Correlation between Neuroretinal Rim and Optic Disc Areas in Normal Melanoderm and Glaucoma Patients. J Fr Ophtalmol. 23 (2000)
3. Bonomi, L., Orzalesi, N.: Glaucoma: Concepts in Evolution. Morphometric and Functional Parameters in the Diagnosis and Management of Glaucoma. Kugler Publications, New York (1991)
4. Fitzgibbon, A., Pilu, M., Fisher, R.B.: Direct Least Square Fitting of Ellipses. IEEE Tran. on Pattern Analysis & Machine Intelligence 21 (1999)
5. Hamilton, A.M.P., Ulbig, M.W., Polkinghorne, P.: Management of Diabetic Retinopathy. (1996)
6. Hsu, W., Pallawala, P.M.D.S., Lee, M.L., Au-Eong, K.G.: The Role of Domain Knowledge in the Detection of Retinal Hard Exudates. IEEE Conf. on Computer Vision and Pattern Recognition (2001)
7. Kass, A., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. Int. Journal of Computer Vision (1988)
8. Lalonde, M., Beaulieu, M., Gagnon, L.: Fast and Robust Optic Disc Detection Using Pyramidal Decomposition and Hausdorff-Based Template Matching. IEEE Trans. on Medical Imaging (2001)
9. Larsen, H.W.: Manual and Color Atlas of the Ocular Fundus (1976)
10. Lee, S., Brady, M.: Optic Disc Boundary Detection. British Machine Vision Conference (1989)
11. Leroy, B., Herlin, I.L., Cohen, L.D.: Multi-Resolution Algorithms for Active Contour Models, Int. Conf. on Analysis and Optimization of Systems (1996)
12. Lewis, A., Knowles, G.: Image Compression Using the 2-D Wavelet. IEEE Trans. on Image Processing (1992)
13. Li, H., Chutatape, O., Automatic Location of Optic Disc in Retinal Images. Int. Conf. on Image Processing (2001)
14. Mendels, F., Heneghan, C., Harper, P.D., Reilly, R.B., Thiran, J-P.: Extraction of the Optic Disc Boundary in Digital Fundus Images. First Joint BMES/EMBS Conf. Serving Humanity, Advancing Technology (1999)
15. Mendels, F., Heneghan, C., Thiran, J-P.: Identification of the Optic Disc Boundary in Retinal Images Using Active Contours. Irish Machine Vision and Image Processing Conf. (1999)
16. Morris, D.T., Donnison, C.: Identifying the Neuro-Retinal Rim Boundary Using Dynamic Contours. Image and Vision Computing 17 (1999)
17. Park, H.W., Schoepflin, T., Kim, Y.: Active Contour Model with Gradient Directional Information: Directional Snake. IEEE Trans. On Circuits and Systems for Video Technology (2001)
18. Sinthanayothin, C., Boyce, J.F., Cook, H.L., Williamson, T.H.: Automated Localization of Optic Disc, Fovea and Retinal Blood Vessels from Digital Color Fundus Images. British Journal of Ophthalmology(1999)
19. Wang, H., Hsu, W., Goh, K.G., Lee, M.L.: An Effective Approach to Detect Lesions in Color Retinal Images. IEEE Conf. on Computer Vision and Pattern Recognition (2000)
20. World Health Organization Fact Sheet No. 138 (2002)

21. Xu, C., Prince, J.L.: Generalized Gradient Vector Flow External Forces for Active Contours. Signal Processing (1998)
22. Xu, C., Prince, J.L.: Snakes, Shapes, and Gradient Vector Flow. IEEE Trans. on Image Processing (1988)
23. Xu, C., Yezzi, A., Prince, J.L.: On the Relationship Between Parametric and Geometric Active Contours. $34^{th}$ Asimolar Conf. on Signals, Systems, and Computers (2000)
24. Yogesan, K., Barry, C.J., Jitskaia, L., Eikelboom, Morgan, W.H., House, P.H., Saarloos, P.P.V.: Software for 3-D Visualization/Analysis of Optic-Disc Images. IEEE Engineering in Medicine and Biology (1999)

# View-Invariant Recognition Using Corresponding Object Fragments

Evgeniy Bart, Evgeny Byvatov, and Shimon Ullman

Department of Computer Science And Applied Mathematics
Weizmann Institute of Science, Rehovot, Israel 76100
{evgeniy.bart, shimon.ullman}@weizmann.ac.il

**Abstract.** We develop a novel approach to view-invariant recognition and apply it to the task of recognizing face images under widely separated viewing directions. Our main contribution is a novel object representation scheme using 'extended fragments' that enables us to achieve a high level of recognition performance and generalization across a wide range of viewing conditions. Extended fragments are equivalence classes of image fragments that represent informative object parts under different viewing conditions. They are extracted automatically from short video sequences during learning. Using this representation, the scheme is unique in its ability to generalize from a single view of a novel object and compensate for a significant change in viewing direction without using 3D information. As a result, novel objects can be recognized from viewing directions from which they were not seen in the past. Experiments demonstrate that the scheme achieves significantly better generalization and recognition performance than previously used methods.

## 1 Introduction

View-invariance refers to the ability of a recognition system to identify an object, such as a face, from any viewing direction, including directions from which the object was not seen in the past. View-invariant recognition is difficult because images of the same object viewed from different directions can be highly dissimilar. The challenge of view-invariant recognition is to correctly identify a novel object based on a limited number of views, from different viewing directions. For example, after seeing a single frontal image of a novel face, the same face has to be recognized when seen in profile.

In the current study we develop a scheme for view-invariant recognition based on the automatic extraction and use of corresponding views of informative object parts. The approach has two main components. First, objects within a class, such as face images, are represented in terms of common 'building blocks', or parts. The parts we use are sub-images, or object fragments, selected automatically from a training set during a learning phase. Second, images of the same part under different viewing directions are grouped together to form a generalized fragment that extends across changes in the viewing direction. (We therefore refer to a set of equivalent fragments as an 'extended' fragment.) The general

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 152–165, 2004.
© Springer-Verlag Berlin Heidelberg 2004

idea is that the equivalence between different object views will be induced by the learned equivalence of the extended fragments. To achieve view invariance, the view of a novel object within a class will be represented in terms of the constituent parts. The appearances of the parts themselves under different conditions are extracted during learning, and this will be used for recognizing the novel object under new viewing conditions. We describe below how such extended fragments are extracted from training images, and how they are used for identifying novel objects seen in one viewing direction, from a different and widely separated direction.

The remainder of the paper is organized as follows. In section 2 we review past approaches to view-invariant recognition. In section 3 our extended fragments representation is introduced, and in section 4 we describe how this representation is incorporated in a recognition scheme. In section 5 we present results obtained by our algorithm and compare them to a popular PCA-based approach. We make additional comparisons and discuss future extensions in section 6.

## 2 A Review of Past Approaches

In this section, we give a brief review of some general approaches to view-invariant recognition.

One possible approach to achieve view invariance is to use features that are by themselves invariant to pose transformations. The basic idea is to identify image-based measures that remain constant as a function of viewing direction, and use them as a signature that identifies an object. One well-known measure is the four-point cross-ratio, but other, more complicated algebraic invariants, have been proposed [1]. Several types of features invariant under arbitrary affine transformations were derived and used for object recognition [2,3], and features that are nearly invariant were derived for more general transformations [4]. One shortcoming of this approach is that it is difficult to find a sufficient number of invariant features for reliable recognition, especially when objects that are similar in overall shape (such as faces) have to be discriminated. Second, many useful features are not by themselves invariant, and consequently their use is excluded in the invariant features framework, in contrast with the extended features approach described below . (See also comparisons in section 3.)

Another general approach is to store multiple views of each object to be recognized, and possibly apply some form of view-interpolation for intermediate views (e.g. [5]). This approach requires multiple views of each novel object, and the interpolation usually requires correspondence between the novel object and a stored view. Such correspondence turned out in practice to be a difficult problem.

Having a full 3D model of an object alleviates the need to store multiple views, since novel views may be generated from such a model. However, obtaining precise 3D models in practice is difficult, and usually requires special measuring equipment (e.g. [6]). Due to this requirement, recognition using 3D data is frequently considered separately from image-based methods. For examples of 3D approaches, see [7,8].

Several methods (elastic graph matching [9], Active Appearance Models [10]) use flexible matching to deal with the deformation caused by changes in pose.

Since small pose changes tend to leave all features visible and only change the distances between them, this approach is able to compensate for small ($10 - 15°$) head rotations. A feature common to such approaches is that they easily compensate for small pose changes, but the performance drops significantly when larger pose changes (e.g. above $45°$) are present.

A popular approach to object recognition in general is based on principal components analysis (PCA). When applied to face recognition, this approach is known as eigen-faces [11]. Several researchers have used PCA to achieve pose invariance. Murase and Nayar [12] acquired images of several objects every four degrees. From these images, they constructed an eigenspace representation for a given object, and used it for recognizing the object in different poses. A limitation of this approach is the need to acquire and store a large number of views for each object. Pentland et al. [13] developed a similar scheme, applied to face images, and using only five views between frontal and profile (inclusive). Performance was good when the view to be recognized was at the same orientation as the previously seen pictures, but dropped quickly when interpolation or extrapolation between views was required.

## 3 The Extended Fragments Approach

Our approach is an extension of object recognition methods in which objects are represented using a set of informative sub-images, called fragments or patches [14,15,16]. These methods are general and can be applied to a wide variety of natural object classes (for example, faces, cars, and animals). In this section we describe briefly the relevant aspects of the fragment-based approaches, discuss their limitations for view-invariance, and outline our extension based on extended fragments. We illustrate the approach using the task of face recognition, but the method is general and can be applied to different object classes.

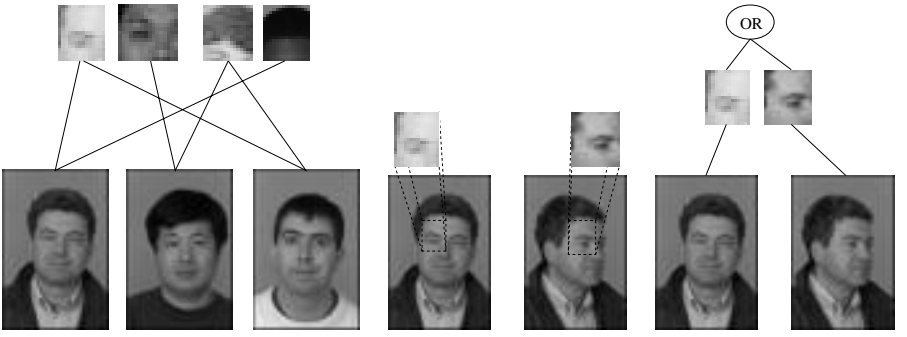
In fragment-based recognition, informative object fragments are extracted during a learning stage. The extraction is based on the measure of mutual information between the fragments and the class they represent. A large set of candidate fragments is evaluated, and a subset of informative fragments is then selected for the recognition process. Informative fragments for face images typically include different types of eyes, mouths, hairlines, etc.

During recognition, this set of fragments is searched for in the target images using the absolute value of normalized cross-correlation, given by

$$NCC(p, f) = \frac{E(p - \overline{p})(f - \overline{f})}{\sigma_p \sigma_f}. \quad (1)$$

Here $f$ is the fragment and $p$ is an image patch of the same size as the fragment. Image patches at all locations are evaluated and the one with the highest correlation is selected. When the correlation exceeds a pre-determined threshold, the fragment is considered present, or *active*, in the image. A schematic illustration of this scheme is presented in Figure 1(a).

Informative fragments have a number of desirable properties [14]. They provide a compact representation of objects or object classes and can be used for




(a) Fragments-based recognition (b) Lack of viewpoint-invariance (c) Extended fragments

**Fig. 1.** Extended object fragments. (a) Schematic illustration of previous fragments-based approaches. Bottom: faces represented in the system. Top: informative fragments used for the representation. Lines connect each face to fragments that are present in the face, as computed by normalized cross-correlation. Novel faces can be detected reliably using a limited set of fragments. (b) Informative fragments are not viewpoint-invariant, for instance, a frontal eye fragment (left) is different from the corresponding side fragment (right). If the detection threshold is set low enough so that the fragment will be active in both images, many spurious detections will occur, and the overall recognition performance will deteriorate. (c) View invariance is obtained by introducing equivalence sets of fragments. Fragments depicting the same face part viewed from different angles are grouped to form an extended fragment. The eye is detected in an image if either the frontal or the profile eye templates are found. This is indicated by the OR attached to the pair of fragments.

efficient and accurate recognition. However, fragments used in previous schemes are not view-invariant. The reason is that objects and object parts look very different under different orientations. As a result, fragments that were active e.g. in a frontal view of a certain face will not be active in side views of the same face. Therefore, the representation by active fragments is not view-invariant. This problem is illustrated in Figure 1(b).

To overcome this problem, we use the fact that the only source of difference between the left and right images in Figure 1(b) is different viewpoint. The face itself consists of the same sub-parts in both images, and we therefore wish to represent the objects in terms of sub-parts and not in terms of view-specific sub-images. The representation using sub-parts will then be view-invariant and will allow invariant recognition.

To represent sub-parts in an invariant manner, one approach has been to use affine-invariant patches [3,2]. This approach works well in some applications (such as wide-baseline matching) that use nearly planar surfaces. However, for non-planar objects, including faces, affine transformations provide a poor appro-

ximation. In our experiments, methods based on affine invariant matching failed entirely at 45° rotation.

In our scheme, invariance of sub-parts was achieved using multiple templates, by grouping together the images of the same object part under different viewing conditions. For example, to represent the 'eye' part in Figure 1(b), the two sub-images of the eye region shown in the figure are grouped together to form an 'extended eye fragment'. Using this extended fragment, the eye is detected in an image if either the frontal or the side eye template are detected. Typically, the frontal template would be found in frontal images and the profile template would be found in profile images. Consequently, at the level of extended fragments, the representation becomes invariant, as illustrated schematically in Figure 1(c).

Note that the scheme uses only multiple-template representation for *object parts*, not for entire objects as was used by previous multiple template algorithms [5,12,13]. This has a number of significant advantages over previous schemes. First, multiple examples of each object are needed only in the training phase, when extended fragments are created. Since extended fragments are both view-invariant and capable of representing novel objects of the same class, view-invariant representation of novel objects is obtained from a single image. This is a significant advantage over previous multiple-views schemes, where many views for each novel objects were required. Second, the extended fragments representation is more efficient in terms of memory than previous multiple-template schemes, because the templates required for fragment representation are much smaller that the entire object images. The reduction in space requirements also reduces matching time, as there is no need to perform matching of a large collection of full-size images. Finally, object parts generalize better to novel viewing conditions than entire objects (see section 6). Therefore, fewer templates per extended fragment will be required to cover a given range of viewing directions compared with matching images of entire objects.

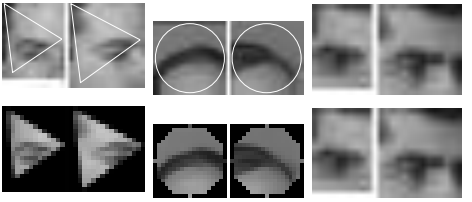
We have implemented the scheme outlined above and applied it to recognize face images from two widely separated viewing directions: frontal and 60° profile, called below a 'side view'. Note that this range is wide enough to undermine schemes that were not specifically designed for view-invaraince, such as [9,10]. As discussed in section 6, generalization of our algorithm to handle any viewing direction is straightforward. The following sections describe in more detail the different stages in the algorithm.

### 3.1 The Extraction of Extended Fragments

In our training, extended fragments were extracted from images of 100 subjects from the FERET database [17]. The images were low-pass filtered and down-sampled to size $60 \times 40$ pixels.

To form extended fragments, the multiple template representation of object parts must be obtained. To deal with two separate views, a set of sub-image pairs must be provided, where in each pair one sub-image will be a view of some face part in frontal orientation and the other sub-image will be a view of the same face part in the side view. For this, correspondence must be established between

(a) A sample training sequence



(b) Example fragments

**Fig. 2.** Illustration of extended fragments selection. (a) A sample training sequence. (b) Some extended fragments that were selected automatically from this and similar training sequences. Each pair constitutes an extended fragment; the left part is the fragment in frontal images, the right part is the same fragment in side images. Top part: sub-images with fragment shapes delineated. Bottom part: the same fragments are shown outside the corresponding sub-images.

face areas in frontal and side images. This is a standard task in computer vision that is similar to optical flow computation.

We used the KLT (Kanade-Lucas-Tomasi) algorithm [18] to automatically establish correspondences between frontal and side views. The KLT algorithm selects points in the initial image that can be tracked reliably, and uses a simple gradient search to follow the selected points in subsequent images. The tracking improves when the differences between successive images are small. Therefore, using short video segments of rotating faces produces more reliable results. We have used in the training stage the images from the FERET database [17]. A training sequence contained three intermediate views in addition to the frontal and size views; an example of such a sequence is shown in Figure 2(a).

After tracking was completed, the intermediate views were discarded. The correspondences obtained by KLT can then be used to associate together the views of the same face part under different poses. This is obtained by selecting a sub-image in one view (e.g. frontal), and using the tracked points to identify the corresponding sub-image from the other (side) view. The sub-images are polygons defined by a subset of matching points. In particular, we have used triangular sub-images defined by corresponding triplets of points. Matching pairs of triangles (one from frontal, another from side view) were grouped together and formed the pool of candidate extended fragments. We have also tried to interpolate between the points tracked by KLT to obtain dense correspondences, and use matching regions of arbitrary shape (Figure 2(b)). However, the difference in performance was only marginal. Therefore, triangular fragments were used throughout most of our tests.

### 3.2 Selection Using Mutual Information and Max-Min Algorithm

During learning, extended fragments were extracted from the 100 training sequences. The number of all possible fragments was about 100000 per sequence;

as a result, learning becomes time-consuming. However, most of these candidate fragments are not informative, and it is possible to reduce the size of the pool based on the fragments size. It was shown in [15] that most informative fragments have intermediate size. In our experiments (see section 6), fragments that were smaller than 6% of the object area or larger than 20% were uninformative. By excluding from consideration candidate fragments outside these size constraints, the number of candidate extended fragments was reduced to about 1000 per training sequence. Since calculating the fragment's size is much simpler than evaluating its mutual information, significant savings in computation were obtained.

Extracting extended fragments from all 100 sequences results in a pool of candidate fragments of size around 100000. This set still contains many redundant or uninformative extended fragments. Therefore, the next stage in the feature extraction process is to select a smaller subset of fragments that are most useful for the task of recognition.

This selection was obtained based on maximizing the information supplied by the extended fragments for view-invariant recognition. The use of mutual information for feature selection is motivated by both theoretical and experimental results. Successful classification reduces the initial uncertainty (entropy) about the class. The classification error is bounded by the residual entropy (Fano's inequality [19]), and this entropy is minimized when $I(C;F)$, the mutual information between the class and the set $F$ of fragments, is maximal. In practice, selecting features based on maximizing mutual information leads to improved classification compared with less informative features [14]. We explain below the procedure for selecting the most informative extended fragments.

This first step of the selection procedure derives for each extended fragment a measure of mutual information. Mutual information between the class $C$ and fragment $F$ is given by

$$I(C;F) = \sum_{c,f} p(C=c, F=f) \log \frac{p(C=c, F=f)}{p(C=c)p(F=f)}. \tag{2}$$

By measuring the frequencies of detecting $F$ inside different classes $c$, we can evaluate the mutual information of a fragment from the training data.

The next step is to select a bank of $n$ fragments $B = \{F_1, \ldots, F_n\}$ with the highest mutual information about the class $C$; $B = \arg\max I(C;B)$. Evaluating the mutual information with respect to the joint distribution of many variables is impractical, therefore some approximation must be used. A natural approach is to use greedy iterative optimization. The selection process is initialized by selecting the extended fragment $F_1$ with the highest mutual information. Fragments are then added one by one, until the gain in mutual information is small, or until a limit on the bank size $n$ is reached. To expand a size-$k$ fragment bank $B = \{F_1, \ldots, F_k\}$ to size $k+1$, a new fragment $F_{k+1}$ must be selected that will add the maximal amount of new information to the bank. The conditional mutual information between $F_{k+1}$ and the class given the current fragment bank must therefore be maximized: $F_{k+1} = \arg\max I(C; F_{k+1}|B)$. Estimating $I(C; F_{k+1}|B)$ still depends on multiple fragments. The term $I(C; F_{k+1}|B)$ can be approximated by $\min_{F_i \in B} I(C; F_{k+1}|F_i)$. This term contains just two fragments and can

be computed efficiently from the training data. The approximation essentially takes into account correlations between pairs of fragments, but not higher order interactions. It makes sure that the new fragment $F_{k+1}$ is informative, and that the information it contributes is not contained in any of the previously selected fragments. The overall algorithm for selection can be summarized as:

$$F_1 = \arg\max_F I(C; F); \tag{3}$$

$$F_{k+1} = \arg\max_F \min_i I(C; F|F_i). \tag{4}$$

The second stage determines the contribution of a fragment $F$ by finding the most similar fragment already selected (this is the min stage) and then selects the new fragment with the largest contribution (the max stage). The full computation is therefore called the max-min selection.

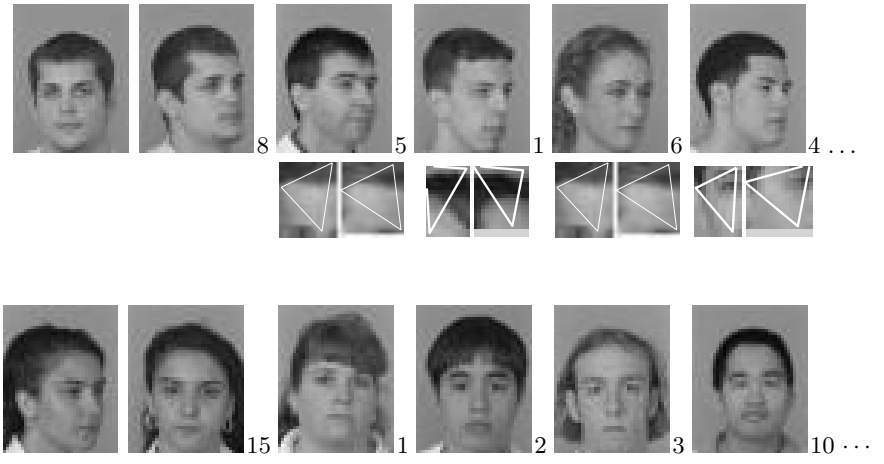



**Fig. 3.** An example of recognition by extended fragments. (a) Top row: a novel frontal face (left image), together with the same face, but at 60° orientation, among distractor faces. Only a few examples are shown, the actual testing set always contained 99 distractors. The side view images are arranged according to their similarity to the target image computed by the extended fragments algorithm. The image selected as the most similar is the correct answer. Below each distractor, one of the extended fragments that helped in the identification task is shown. Each of these fragments was detected either only in the frontal face, or only in the distractor side view above the fragment, providing evidence that the two faces are different. The numbers next to each face show its rank as given by the view-based PCA scheme. (b) Same as (a), without the fragments shown. The test faces were frontal in this case and the target face was at 60°.

During recognition, fragment detection is performed by computing the absolute value of the normalized cross-correlation at every image location, and

selecting the location with the highest correlation. The maximal correlation, which is a continuous value in the range $[0, 1]$, can be used in the recognition process. We used, however, a simplified scheme in which the feature value was binarized. A fragment was considered to be present, and have the value of 1 in a given image, if its maximal correlation in the image was above a pre-determined threshold. If the maximal correlation was below the threshold, the fragment was assigned value of 0. (Fragments whose activation is above the threshold are also called 'active' below, and those with activation below the threshold are called 'inactive'.) An optimal threshold was selected automatically for each fragment in such a way as to maximize the fragment's mutual information with the class: $\theta = \arg\max I(C; F_\theta)$. Here $F_\theta$ is the fragment $F$ detected with threshold $\theta$. Since extended fragments consist of several individual fragments (two in our case), each has a separate threshold. These thresholds can be determined by a straightforward search procedure.

Examples of extended fragments selected automatically by the algorithm are shown in Figure 2(b).

## 4 Recognition Using Extended Fragments

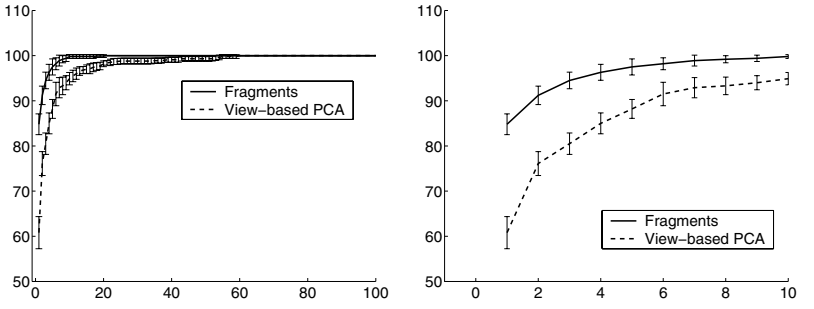
In performing recognition, the system is given a single image of a novel face, for example, in frontal view. It is also presented with a gallery of side views of different faces. The task is to identify the side view image from the gallery that corresponds to the frontal view.

Given the extended fragments representation, the recognition procedure is straightforward. The novel image is represented by the activation pattern of the fragment bank. This is a binary vector that specifies which of the extended fragments were active in the image. Similarly, activation patterns of the gallery images are known. SVM classifier was used to identify the side activation pattern that corresponds to the given frontal activation pattern. An example of the scheme applied to target and test images is shown in Figure 3.

## 5 Results

In this section we summarize the results obtained by the method presented above and compare them with other methods. The results were obtained as follows. The database images were divided into a training and a testing set (several random partitions were tried in every experiment). In the training phase, images of 100 individuals were used. For each individual the data set contained 5 images in the orientations shown in Figure 2(a). This set of images was used to select 1000 extended fragments and their optimal thresholds as described in section 3.2 and train the SVM classifier.

In the testing phase, the algorithm was given a novel frontal view, called the target view. (All individuals in the testing and training phases were different.) The task was then to identify the side view of the individual shown in the novel frontal view. The algorithm was presented with a set (called the 'test set') of 100 side views of different people, one of which was of the same individual shown in




(a) Recognition performance (b) Same as (a), magnified

**Fig. 4.** Recognition results and some comparisons. The graph value at $X = k$ shows the percentage of trials for which the correct classification was among the top $k$ choices. Bars show standard deviation. (a) Comparison of extended fragments with PCA. (b) Initial portion of the plot in (a), magnified.

the frontal view. The algorithm ranked these pictures by their similarity to the frontal view using the extended features as described above. When the top ranked picture corresponded to the target view, the algorithm correctly recognized the individual.

We present our results using CMC (cumulative match characteristics) curves. A CMC curve value at point $X = k$ shows the percentage of trials for which the correct match was among the top $k$ matches produced by the algorithm. Typically, the interesting region of the curve comprises the several initial points; in particular, the point $x = 1$ on the curve corresponds to the frequency at which the correct view was ranked first among the 100 views, i.e. was correctly recognized.

The results of our scheme are shown in Figure 4. As can be seen, side views of a novel person were identified correctly in about 85% of the cases following the presentation of a single frontal image. In order to compare this performance to previous schemes, we implemented the view-invariant PCA scheme of Pentland et al. [13], which is one of the most successful and widely used face recognition approaches. Our implementation was identical to the scheme as described in [13]. PCA performance was calculated under exactly the same conditions as used for our algorithm, i.e. we trained PCA on the same training images and tested recognition on the same images. Figure 4(a) shows the results of the comparison. As can be seen from the figure, this method identifies the person correctly in 60% of the cases. The plots in Figure 4 show the marked advantage of the present algorithm over PCA (the differences are significant at $p < 0.01$, $\chi^2$ test).

A recognized weakness of the PCA method is that it requires precise alignment of the images. In contrast, our algorithm can tolerate significant errors in alignment. We have tested the sensitivity of both algorithms to alignment precision. To test the sensitivity of the extended fragments scheme, we fixed one

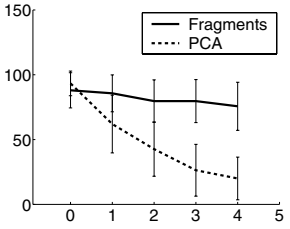



**Fig. 5.** The effect of misalignment on recognition performance. Percent correct recognition as a function of misalignment magnitude in pixels for extended fragments and view-invariant PCA.

(frontal) part of each fragment, and shifted the other (side) part in a random direction by progressively larger amounts from its correct position. This created a controlled error in the location of corresponding fragments. We tested the recognition performance as a function of the correspondence error. The results were compared with a similar test applied to the PCA method, where the images were not precisely aligned, but had a systematic misalignment error. Figure 5 shows the performance of the schemes as a function of the amount of misalignment. (These tests were performed on a new database and with a smaller test set, therefore the results are not on the same scale as in previous figures.) The task was to recognize one out of five faces. Note that for four-pixel shifts, corresponding on average to 12% of the face size, PCA performance reduces to chance level. In many schemes, image misalignment during learning is a significant potential source of errors. As seen in the figure, extended fragments are significantly more robust than PCA to misalignments in the learning stage.

## 6 Discussion

We described in this work a general approach to view-invariant object recognition. The approach is based on a novel type of features, which are equivalence sets of corresponding sub-images, called extended fragments. The features are class-based and are applicable to many natural object classes. In particular, we have applied the approach to cars and animals with similar results.

Despite the large number of extended fragments used, time requirements of the recognition stage are quite reasonable. In our tests, 1500 recognition attempts using 1000 fragments took 2–3 seconds (without optimizing the code for efficiency). Time requirements of the learning stage are more significant, but learning is performed off-line.

One potential limitation of the extended fragments representation is that due to the local nature of the fragments, they might be detected in face images under inconsistent orientations (e.g. frontal fragment would be detected in profile view), which might lead to a decrease in performance. However, in experiments where fragments were restricted to be detected only in the relevant orientations, no performance improvement was observed. The implication is that reliable

recognition can be based on the activation of the features themselves, without explicitly testing for view consistency between different features.

In the feature selection process, the size of preferred features was not fixed, but free to change within a general range (set in the simulations between 6–20% of object size). In other schemes, features are often required to be of a fixed size, with some schemes using small local features, and others using global object features. It therefore became of interest to test the size of the most informative features, and to compare the results with alternative approaches.

Previous studies [15] have shown that useful fragments are typically of intermediate size. To investigate this further, we tested the effect of feature size on view-invariant recognition, using a new database of size 50; 40 images for training and 10 for testing. This size of the database allowed selecting extended fragments of various sizes and comparing their performance. We found that when the extended fragments were of size between 6% and 20% of the object area, the test faces were correctly recognized in $89 \pm 10\%$ of the cases. When the selected fragments were constrained to be smaller (below 3% of the object size), the performance dropped to $67 \pm 12\%$, and when the fragments were constrained to be large (above 20%), the performance dropped to $75 \pm 10\%$. Both results were highly significant ($t$ test, $p < 0.01$). This can be contrasted with approaches such as [20,21], that use small local features, or [11,13], where the features are global. The optimal features extracted were also found to vary in size, to represent object features of different dimensions. This can be contrasted with approaches such as [22], where the features are constrained to have a fixed size. This flexibility in feature size is important for maximizing the fragments mutual information and classification performance.

In our testing, features were extracted for two orientations – frontal and 60° side view. A complete recognition model should be able, however, to handle a full range of orientations (from left to right profile and at different elevations). As mentioned above, the scheme can be extended to handle a full range of orientations by including views from a set of representative viewing directions in each extended fragment. Results of a preliminary experiment suggest that 15 views are sufficient to cover the entire range of views, or nine views if the bilateral symmetry of the face is used. These requirements can be compared with typical view interpolation schemes such as [5]. This scheme requires the use of 15 views to achieve recognition within a restricted range of $-30°$ to $30°$ horizontally and $-20°$ to $20°$ vertically. More importantly, it requires all 15 views for each novel face. In contrast, the extended fragments scheme requires 15 views for training only; in testing, recognition of novel faces is performed from a single view.

The proposed approach can be extended to handle sources of variability such as illumination and facial expression. This can be performed within the general framework by adding the necessary templates to each extended fragment. There are indications [23] that compensating for illumination changes will be possible using a reasonable number of templates. Facial expressions often involve a limited area of the face, and therefore affect only a small number of fragments. The full size of equivalence sets of extended fragments required to perform unconstrained recognition is a subject for future work.

**Acknowledgments.** This research was supported in part by the Moross Laboratory at the Weizmann Institute of Science. Portions of the research in this paper use the FERET database of facial images collected under the FERET program [17].

## References

1. Mundy, J., Zisserman, A.: Geometric Invariance in Computer Vision. The MIT press (1992)
2. Tuytelaars, T., Gool, L.V.: Wide baseline stereo matching based on local, affinely invariant regions. In: British Machine Vision Conference. (2000) 412–425
3. Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Proceedings of the European Conference on Computer Vision. (2002) 128–142
4. Wallraven, C., Bülthoff, H.H.: Automatic acquisition of exemplar-based representations for recognition from image sequences. In: CVPR 2001 - Workshop on Models vs. Exemplars. (2001)
5. Beymer, D.J.: Face recognition under varying pose. Technical Report AIM-1461, MIT Artificial Intelligence Lab (1993)
6. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In Rockwood, A., ed.: Siggraph 1999, Computer Graphics Proceedings, Los Angeles, Addison Wesley Longman (1999) 187–194
7. Nagashima, Y., Agawa, H., Kishino, F.: 3D face model reproduction method using multi view images. Proceedings of the SPIE, Visual Communications and Image Processing '91 **1606** (1991) 566–573
8. Lowe, D.G.: Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence **31** (1987) 355–395
9. Wiskott, L., Fellous, J.M., Krüger, N., von der Malsburg, C.: Face recognition by elastic bunch graph matching. In Jain, L.C., Halici, U., Hayashi, I., Lee, S.B., eds.: Intelligent Biometric Techniques in Fingerprint and Face Recognition. CRC Press (1999) 355–396
10. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. In: Proceedings of the European Conference on Computer Vision. (1998) 484–498
11. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience **3** (1991) 71–86
12. Murase, H., Nayar, S.: Visual learning and recognition of 3-d objects from appearance. International Journal of Computer Vision **14** (1995) 5–24
13. Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA (1994)
14. Sali, E., Ullman, S.: Combining class-specific fragments for object recognition. In: British Machine Vision Conference. (1999) 203–213
15. Ullman, S., Vidal-Naquet, M., Sali, E.: Visual features of intermediate complexity and their use in classification. Nature Neuroscience **5** (2002) 682–687
16. Agarwal, S., Roth, D.: Learning a sparse representation for object detection. In: Proceedings of the European Conference on Computer Vision. (2002) 113–127
17. Phillips, P.J., Wechsler, H., Huang, J., Rauss, P.: The FERET database and evaluation procedure for face recognition algorithms. Image and Vision Computing **16** (1998) 295–306

18. Tomasi, C., Kanade, T.: Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University (1991)
19. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley (1991)
20. Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. Neural Computation **9** (1997) 1545–1588
21. Mel, B.W.: SEEMORE: Combining color, shape and texture histogramming in a neurally-inspired approach to visual object recognition. Neural Computation **9** (1997) 777–804
22. Weber, M., Welling, M., Perona, P.: Towards automatic discovery of object categories. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2002) 101–109
23. Basri, R., Jacobs, D.W.: Lambertian reflectance and linear subspaces. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 218–233

# Variational Pairing of Image Segmentation and Blind Restoration

Leah Bar[1], Nir Sochen[2], and Nahum Kiryati[1]

[1] School of Electrical Engineering
[2] Dept. of Applied Mathematics
Tel Aviv University, Tel Aviv 69978, Israel

**Abstract.** Segmentation and blind restoration are both classical problems, that are known to be difficult and have attracted major research efforts. This paper shows that the two problems are tightly coupled and can be successfully solved together. Mutual support of the segmentation and blind restoration processes within a joint variational framework is theoretically motivated, and validated by successful experimental results. The proposed variational method integrates Mumford-Shah segmentation with parametric blur-kernel recovery and image deconvolution. The functional is formulated using the $\Gamma$-convergence approximation and is iteratively optimized via the alternate minimization method. While the major novelty of this work is in the unified solution of the segmentation and *blind* restoration problems, the important special case of *known* blur is also considered and promising results are obtained.

## 1 Introduction

Image analysis systems usually operate on blurred and noisy images. The standard model $g = h * f + n$ is applicable to a large variety of image corruption processes that are encountered in practice. Here $h$ represents an (often unknown) space-invariant blur kernel (point spread function), $n$ is the noise and $f$ is an ideal version of the observed image $g$.

The two following problems are at the basis of successful image analysis. (1) Can we segment $g$ in agreement with the structure of $f$? (2) Can we estimate the blur kernel $h$ and recover $f$? Segmentation and blind image restoration are both classical problems, that are known to be difficult and have attracted major research efforts, see e.g. [16,3,8,9].

Had the correct segmentation of the image been known, blind image restoration would have been facilitated. Clearly, the blur kernel could have then been estimated based on the smoothed profiles of the known edges. Furthermore, denoising could have been applied to the segments without over-smoothing the edges. Conversely, had adequate blind image restoration been accomplished, successful segmentation would have been much easier to achieve. Segmentation and blind image restoration are therefore tightly coupled tasks: the solution of either problem would become fairly straightforward given that of the other.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 166–177, 2004.
© Springer-Verlag Berlin Heidelberg 2004

This paper presents an integrated framework for simultaneous segmentation and blind restoration. As will be seen, strong arguments exist in favor of constraining the recovered blur kernel to parameterized function classes, see e.g. [4]. Our approach is presented in the context of the fundamentally important model of isotropic Gaussian blur, parameterized by its (unknown) width.

## 2 Fundamentals

### 2.1 Segmentation

The difficulty of image segmentation is well known. Successful segmentation requires top-down flow of models, concepts and a priori knowledge in addition to the image data itself. In their segmentation method, Mumford and Shah [10] introduced top-down information via the preference for piecewise-smooth segments separated by well-behaved contours. Formally, they proposed to minimize a functional that includes a fidelity term, a piecewise-smoothness term, and an edge integration term:

$$\mathcal{F}(f,K) = \frac{1}{2}\int_{\Omega}(f-g)^2 dA \; + \\ + \beta \int_{\Omega\setminus K} |\nabla f|^2 dA \; + \; \alpha \int_K d\sigma \tag{1}$$

Here $K$ denotes the edge set and $\int_K d\sigma$ is the total edge length. The coefficients $\alpha$ and $\beta$ are positive regularization constants. The primary difficulty in the minimization process is the presence of the unknown discontinuity set $K$ in the integration domains.

The $\Gamma$-convergence framework approximates an irregular functional $\mathcal{F}(f,K)$ by a sequence $\mathcal{F}_\epsilon(f,K)$ of regular functionals such that

$$\lim_{\epsilon\to 0} \mathcal{F}_\epsilon(f,K) = \mathcal{F}(f,K)$$

and the minimizers of $\mathcal{F}_\epsilon$ approximate the minimizers of $\mathcal{F}$. Ambrosio and Tortorelli [1] applied this approximation to the Mumford-Shah functional, and represented the edge set by a characteristic function $(1-\chi_K)$ which is approximated by an auxiliary function $v$, i.e., $v(x)\approx 0$ if $x\in K$ and $v(x)\approx 1$ otherwise. The functional thus takes the form

$$\mathcal{F}_\epsilon(f,v) = \frac{1}{2}\int_{\Omega}(f-g)^2 dA \; + \; \beta\int_{\Omega} v^2|\nabla f|^2 dA \; + \\ + \alpha \int_{\Omega}\left(\epsilon|\nabla v|^2 \; + \; \frac{1}{4\epsilon}(v-1)^2\right) dA. \tag{2}$$

Richardson and Mitter [12] extended this formulation to a wider class of functionals. Discretization of the Mumford-Shah functional and its $\Gamma$-convergence approximation is considered in [5]. Additional perspectives on variational segmentation can be found in Vese and Chan [19] and in Samson *et al* [15].

Simultaneous segmentation and restoration of a blurred and noisy image has recently been presented in [7]. A variant of the Mumford-Shah functional was approached from a curve evolution perspective. In that work, the discontinuity set is limited to an isolated closed curve in the image and the blurring kernel $h$ is assumed to be a priori known.

### 2.2 Restoration

Restoration of a blurred and noisy image is difficult even if the blur kernel $h$ is known. Formally, finding $f$ that minimizes

$$\|h * f - g\|^2_{L^2(\Omega)} \tag{3}$$

is an ill-posed inverse problem: small perturbations in the data may produce unbounded variations in the solution. In Tikhonov regularization [18], a smoothing term $\int_\Omega |\nabla f|^2 dA$ is added to the fidelity functional (3). In image restoration, Tikhonov regularization leads to over-smoothing and loss of important edge information. For better edge preservation, the Total Variation approach [13,14] replaces $L_2$ smoothing by $L_1$ smoothing. The functional to be minimized is thus

$$\mathcal{F}(f,h) \;=\; \frac{1}{2}\|h * f - g\|^2_{L^2(\Omega)} \;+\; \beta \int_\Omega |\nabla f| dA \;. \tag{4}$$

This nonlinear optimization problem can be approached via the half-quadratic minimization technique [2]. An efficient alternative approach, based on the lagged diffusivity fixed point scheme and conjugate gradients iterations, was suggested by Vogel and Oman [20].

Image restoration becomes even more difficult if the blur kernel $h$ is not known in advance. In addition to being ill-posed with respect to the image, the blind restoration problem is ill-posed in the kernel as well. To illustrate one aspect of this additional ambiguity, suppose that $h$ represents isotropic Gaussian blur, with variance $\sigma^2 = 2t$:

$$h_t = \frac{1}{4\pi t} e^{-\frac{x^2+y^2}{4t}} \;.$$

The convolution of two Gaussian kernels is a Gaussian kernel, the variance of which is the sum of the two originating variances:

$$h_{t_1} * h_{t_2} = h_{t_1+t_2} \;. \tag{5}$$

Assume that the true $t$ of the blur kernel is $t = T$, so $g = h_T * f$ . The fidelity term (3) is obviously minimized by $f$ and $h_T$. However, according to Eq. 5, $g$ can also be expressed as

$$g = h_{t_1} * h_{t_2} * f \qquad \forall (t_1 + t_2) = T \;\;.$$

Therefore, an alternative hypothesis, that the original image was $h_{t_2} * f$ and the blur kernel was $h_{t_1}$, minimizes the fidelity term just as well. This exemplifies a

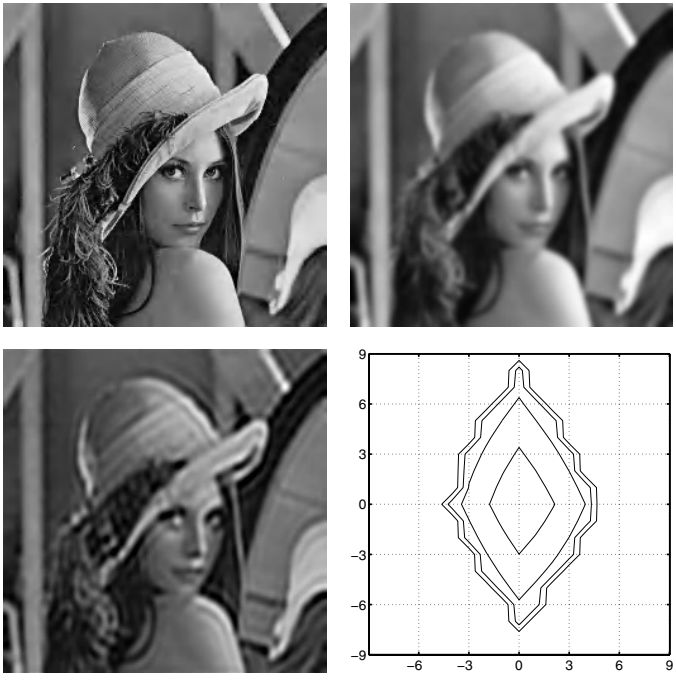



**Fig. 1.** Blind image restoration using the method of [6]. *Top-left:* Original. *Top-right:* Blurred using an isotropic Gaussian kernel ($\sigma = 2.1$). *Bottom-left:* Recovered image. *Bottom-right:* Reconstructed kernel.

fundamental ambiguity in the division of the apparent blur between the recovered image and the blur kernel, i.e., that the scene itself might be blurred. For meaningful image restoration, this hypothesis must be rejected and the largest possible blur should be associated with the blur kernel. It can be achieved by adding a kernel-smoothness term to the functional.

Blind image restoration with joint recovery of the image and the kernel, and regularization of both, was presented by You and Kaveh [21], followed by Chan and Wong [6]. Chan and Wong suggested to minimize a functional consisting of a fidelity term and total variation ($L_1$ norm) regularization for both the image and the kernel:

$$\mathcal{F}(f,h) \;=\; \frac{1}{2}\|h * f - g\|^2_{L^2(\Omega)} \;+\; \alpha_1 \int_\Omega |\nabla f| dA \;+\; \alpha_2 \int_\Omega |\nabla h| dA \;. \qquad (6)$$

Much can be learned about the blind image restoration problem by studying the characteristics and performance of this algorithm. Consider the images shown in Fig. 1. An original image (upper left) is degraded by isotropic Gaussian blur with $\sigma = 2.1$ (top-right). Applying the algorithm of [6] (with $\alpha_1 = 10^{-4}$ and $\alpha_2 = 10^{-4}$) yields a recovered image (bottom-left) and an estimated kernel

(bottom-right). It can be seen that the identification of the kernel is inadequate, and that the image restoration is sensitive to the kernel recovery error.

To obtain deeper understanding of these phenomena, we plugged the original image $f$ and the degraded image $g$ into the functional (6), and carried out minimization only with respect to $h$. The outcome was similar to the kernel shown in Fig. 1 (bottom-right). This demonstrates an excessive dependence of the recovered kernel on the image characteristics. At the source of this problem is the aspiration for general kernel recovery: the algorithm of [6] imposes only mild constraints on the shape of the reconstructed kernel. This allows the distribution of edge directions in the image to have an influence on the shape of the recovered kernel, via the trade-off between the fidelity and kernel smoothness terms. For additional insight see Fig. 2.

Facing the ill-posedness of blind restoration with a general kernel, two approaches can be taken. One is to add relevant data; the other is to constrain the solution. Recent studies have adopted one of these two approaches, or both. In [17], the blind restoration problem is considered within a multichannel framework, where *several* input images can be available.

In many practical situations, the blurring kernel can be modeled by the physics/optics of the imaging device and the set-up. The blurring kernel can then be constrained and described as a member in a class of parametric functions. This constraint was exploited in the direct blind deconvolution algorithm of [4]. In [11], additional relevant data was introduced via learning of similar images and the blur kernel was assumed to be Gaussian.

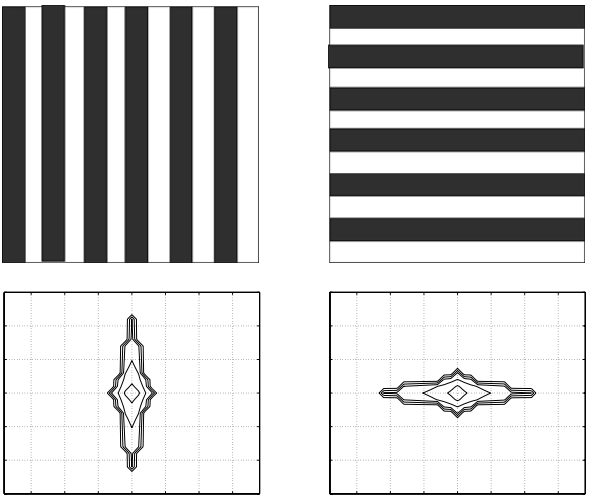


**Fig. 2.** Experimental demonstration of the dependence of the recovered kernel on the image characteristics in [6]. Each of the two synthetic bar images (top-row) was smoothed using an isotropic Gaussian kernel, and forwarded to the blind restoration algorithm of [6] (Eq. 6). The respective recovered kernels are shown in the bottom row.

## 3 Coupling Segmentation with Blind Restoration

The observation, discussed in the introduction, that segmentation and blind restoration can be mutually supporting, is the fundamental motivation for this work. We present an algorithm, based on functional minimization, that iteratively alternates between segmentation, blur identification and restoration.

Concerning the sensitivity of general kernel recovery, we observe that large classes of practical imaging problems are compatible with reasonable constraints on the blur model. For example, Carasso [4] described the association of the Gaussian case with diverse applications such as undersea imaging, nuclear medicine, computed tomography scanners, and ultrasonic imaging in nondestructive testing.

In this work we integrate Mumford-Shah segmentation with blind deconvolution of isotropic Gaussian blur. This is accomplished by extending the Mumford-Shah functional and applying the $\Gamma$-convergence approximation as described in [1]. The observed image $g$ is modeled as $g = h_\sigma * f + n$ where $h_\sigma$ is an isotropic Gaussian kernel parameterized by its width $\sigma$, and $n$ is white Gaussian noise. The objective functional used is

$$\mathcal{F}_\epsilon(f, h_\sigma, v) = \frac{1}{2}\int_\Omega (h_\sigma * f - g)^2 dA \; + \; \beta \int_\Omega v^2 |\nabla f|^2 dA \quad + \\ + \quad \alpha \int_\Omega \left( \epsilon |\nabla v|^2 \; + \; \frac{(v-1)^2}{4\epsilon} \right) dA + \gamma \int_\Omega |\nabla h_\sigma|^2 dA \qquad (7)$$

The functional depends on the functions $f$ (ideal image) and $v$ (edge integration map), and on the width parameter $\sigma$ of the blur kernel $h_\sigma$. The first three terms are similar to the $\Gamma$-convergence formulation of the Mumford-Shah functional, as in (2). The difference is in the replacement of $f$ in the fidelity term by the degradation model $h_\sigma * f$. The last term stands for the regularization of the kernel, necessary to resolve the fundamental ambiguity in the division of the apparent blur between the recovered image and the blur kernel. In the sequel it is assumed that the image domain $\Omega$ is a rectangle in $\mathbb{R}^2$ and that image intensities are normalized to the range $[0, 1]$.

Minimization with respect to $f$ and $v$ is carried out using the Euler-Lagrange equations (8) and (9). The differentiation by $\sigma$ (10) minimizes the functional with respect to that parameter.

$$\frac{\delta \mathcal{F}_\epsilon}{\delta v} = 2\,\beta\, v\, |\nabla f|^2 + \alpha \left( \frac{v-1}{2\epsilon} \right) - 2\,\epsilon\,\alpha\,\nabla^2 v = 0 \qquad (8)$$

$$\frac{\delta \mathcal{F}_\epsilon}{\delta f} = (h_\sigma * f - g) * h_\sigma(-x, -y) - 2\beta\,\mathrm{Div}(v^2\,\nabla f) = 0 \qquad (9)$$

$$\frac{\partial \mathcal{F}_\epsilon}{\partial \sigma} = \int_\Omega \left[ (h_\sigma * f - g) \left( \frac{\partial h_\sigma}{\partial \sigma} * f \right) + \gamma \frac{2h_\sigma^2}{\sigma^5} \left( \frac{x^2 + y^2}{\sigma^2} - 4 \right) \right] dA = 0 \qquad (10)$$

Studying the objective functional (7), it can be seen that it is convex and lower bounded with respect to any one of the functions $f$, $v$ or $h_\sigma$ if the other two

functions are fixed. For example, given $v$ and $\sigma$, $\mathcal{F}_\epsilon$ is convex and lower bounded in $f$. Therefore, following [6], the alternate minimization (AM) approach can be applied: in each step of the iterative procedure we minimize with respect to one function and keep the other two fixed. The discretization scheme used was CCFD (cell-centered finite difference) [20]. This leads to the following algorithm:

`Initialization:` $f = g, \quad \sigma = \varepsilon_1, \quad v = 1, \quad \sigma_{prev} \gg 1$
`while` $(|\sigma_{prev} - \sigma| > \varepsilon_2)$ `repeat`

`1. Solve the Helmholtz equation for` $v$

$$(2\beta\,|\nabla f|^2 + \frac{\alpha}{2\epsilon} - 2\,\alpha\,\epsilon\,\nabla^2)\,v = \frac{\alpha}{2\epsilon}$$

`2. Solve the following linear system for` $f$

$$(h_\sigma * f - g) * h_\sigma(-x, -y) - 2\beta\,\mathrm{Div}(v^2\,\nabla f) = 0$$

`3. Set` $\sigma_{prev} = \sigma$`, and find` $\sigma$ `such that (Eq. 10)`

$$\frac{\partial F_\epsilon}{\partial \sigma} = 0$$

Here $\varepsilon_1$ and $\varepsilon_2$ are small positive constants. Both steps 1 and 2 of the algorithm call for a solution of a system of linear equations. Step 1 was implemented using the Generalized Minimal Residual (GMRES) algorithm. In step 2, a symmetric positive definite operator is applied to $f(x, y)$. Implementation was therefore via the Conjugate Gradients method. In step 3, the derivative of the functional with respect to $\sigma$ was analytically determined, and its zero crossing was found using the bisection method. All convolution procedures were performed in the Fourier Transform domain. The algorithm was implemented in MATLAB environment.

## 4 Special Case: Known Blur Kernel

If the blur kernel is known, the restriction to Gaussian kernels is no longer necessary. In this case, the kernel-regularization term in the objective functional (7) can be omitted. Consequently, the algorithm can be simplified by skipping step 3 and replacing the stopping criterion by a simple convergence measure.

The resulting algorithm for coupled segmentation and image restoration is fast, robust and stable. Unlike [7], the discontinuity set is not restricted to isolated closed contours. Its performance is exemplified in Fig. 3. The top-left image is a blurred and slightly noisy version of the original $256 \times 256$ *Lena* image (not shown). The blur kernel was a pill-box of radius 3.3. The top-right image is the reconstruction obtained using the Matlab's Image Processing Toolbox adaptation of the Lucy-Richardson algorithm (`deconvlucy`). The bottom-left image is the outcome of the proposed method; the bottom-right image shows the associated edge map $v$ determined by the algorithm ($\beta = 1$, $\alpha = 10^{-8}$, $\epsilon = 10^{-5}$, 4 iterations). Computing time was 2 minutes in interpreted MATLAB on a 2GHz PC. The superiority of the suggested method is clear.

**Fig. 3.** The case of a known blur kernel. *Top-left:* Corrupted image. *Top-right:* Restoration using the Lucy-Richardson algorithm (MATLAB: `deconvlucy`). *Bottom-left:* Restoration using the suggested method. *Bottom-right:* Edge map produced by the suggested method.

## 5 Results: Segmentation with Blind Restoration

Consider the example shown in Fig. 4. The top-left image was obtained by blurring the original $256 \times 256$ *Lena* image (not shown) with a Gaussian kernel with $\sigma = 2.1$. The proposed method for segmentation and blind restoration was applied, with $\beta = 1$, $\gamma = 50$, $\alpha = 10^{-7}$ and $\epsilon = 10^{-4}$. The initial value of $\sigma$ was $\varepsilon_1 = 0.5$ and the convergence tolerance was taken as $\varepsilon_2 = 0.001$. Convergence was achieved after 24 iterations; the unknown width of the blur kernel was estimated to be $\sigma = 2.05$, which is in pleasing agreement with its true value. The reconstructed image is shown top-right, and the associated edge map is presented at the bottom of Fig. 4. Compare to Fig. 1.

The top-left image in Fig. 5 is a $200 \times 200$ gray level image, synthetically blurred by an isotropic Gaussian kernel ($\sigma = 2.1$) and additive white Gaussian noise (SNR=44dB). Restoration using [6] ($\alpha_1 = 10^{-4}$, $\alpha_2 = 10^{-4}$) is shown top-right. The reconstruction using the method suggested in this paper ($\beta = 1$, $\alpha = 10^{-6}$, $\gamma = 20$, $\epsilon = 10^{-3}$) is shown bottom-left, and the associated edge map $v$ is shown bottom-right. The number of iterations to convergence was 18, and the estimated width of the blur kernel was 1.9. The convergence process is illustrated in Fig. 6.

**Fig. 4.** Segmentation and blind restoration. *Top-left:* Blurred image ($\sigma = 2.1$). *Top-right:* Blind reconstruction using the proposed algorithm. *Bottom:* Edge map $v$ produced by the suggested method. Compare to Fig. 1.

The last example refers to actual defocus blur. The top-left image in Fig. 7 is a $200 \times 200$ rescaled part of an image obtained with deliberate defocus blur, using a Canon VC-C1 PTZ video communication camera and SGI O2 analog video acquisition hardware. The shape and size of the actual defocus blur were not known to us; they certainly deviate from the isotropic Gaussian model. The top-right image in Fig. 7 shows the blind restoration result of [6] ($\alpha_1 = 10^{-4}$, $\alpha_2 = 10^{-6}$). At the bottom-left is the reconstruction using the method suggested in this paper ($\beta = 1$, $\alpha = 10^{-2}$, $\gamma = 100$, $\epsilon = 0.1$). Convergence was achieved within 7 iterations, and the blur kernel width was estimated to be 1.68. The edge map $v$ is shown bottom-right. The quality of this result demonstrates the applicability of the proposed method to real images and its robustness to reasonable deviations from the Gaussian case. In all our experiments $\beta = 1$, and the best value of $\gamma$, controlling the deconvolution level, was in the range $20 \leq \gamma \leq 100$. The values of $\epsilon$ and $\alpha$ had to be increased in the presence of noise.

## 6 Discussion

This paper validates the hypothesis that the challenging tasks of image segmentation and blind restoration are tightly coupled. Mutual support of the seg-

**Fig. 5.** Segmentation and blind restoration. *Top-left:* Blurred ($\sigma = 2.1$) image with slight additive noise. *Top-right:* Restoration using the method of [6]. *Bottom-left:* Restoration using the suggested method. *Bottom-right:* Edge map produced by the suggested method.

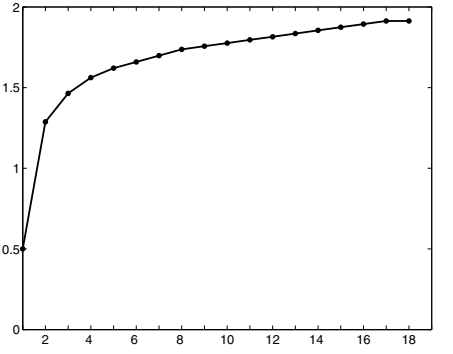



**Fig. 6.** The convergence of the estimated width $\sigma$ of the blur kernel as a function of the iteration number in the blind recovery of the coin image.

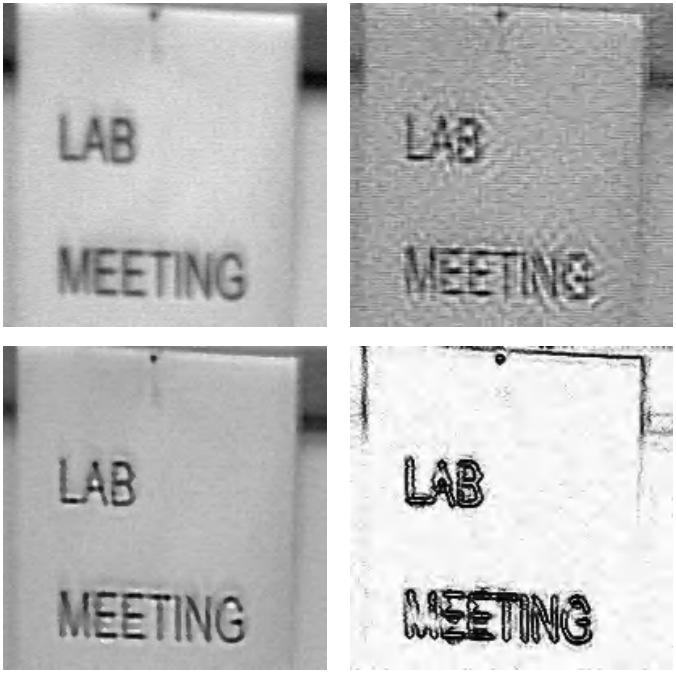



**Fig. 7.** Segmentation and blind restoration of unknown defocus blur. *Top-left:* Blurred image. *Top-right:* Restoration using the method of [6]. *Bottom-left:* Restoration using the suggested method. *Bottom-right:* Edge map produced by the suggested method.

mentation and blind restoration processes within an integrative framework is demonstrated.

Inverse problems in image analysis are difficult and often ill-posed. This means that searching for the solution in the largest possible space is not always the best strategy. A-priori knowledge should be used, wherever possible, to limit the search and constrain the solution. In the context of pure blind restoration, Carasso [4] analyzes previous approaches and presents convincing arguments in favor of restricting the class of blurs.

Along these lines, in this paper the blur kernels are constrained to the class of isotropic Gaussians parameterized by their width. This is a sound approximation of physical kernels encountered in diverse contexts [4]. The advantages brought by this restriction are well-demonstrated in the experimental results that we provide. We plan to extend this approach to parametric kernel classes to which the Gaussian approximation is inadequate, in particular, motion blur.

While the major novelty in this work is in the unified solution of the segmentation and *blind* restoration problems, we have obtained valuable results also in the case of *known* blur, see Fig. 3 (bottom-left). Note that if the blur is known, the restriction to the Gaussian case is no longer necessary.

## References


1. L. Ambrosio and V.M. Tortorelli, "Approximation of Functionals Depending on Jumps by Elliptic Functionals via $\Gamma$-Convergence", *Communications on Pure and Applied Mathematics*, Vol. XLIII, pp. 999-1036, 1990.
2. G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing*, Springer, New York, 2002.
3. M. Banham and A. Katsaggelos, "Digital Image Restoration", *IEEE Signal Processing Mag.*, Vol. 14, pp. 24-41, 1997.
4. A. S. Carasso, "Direct Blind Deconvolution", *SIAM J. Applied Math.*, Vol. 61, pp. 1980-2007, 2001.
5. A. Chambolle, "Image Segmentation by Variational Methods: Mumford and Shah functional, and the Discrete Approximation", *SIAM Journal of Applied Mathematics*, Vol. 55, pp. 827-863, 1995.
6. T. Chan and C. Wong, "Total Variation Blind Deconvolution", *IEEE Trans. Image Processing*, Vol. 7, pp. 370-375, 1998.
7. J. Kim, A. Tsai, M. Cetin and A.S. Willsky, "A Curve Evolution-based Variational Approach to Simultaneous Image Restoration and Segmentation", *Proc. IEEE ICIP*, Vol. 1, pp. 109-112, 2002.
8. D. Kundur and D. Hatzinakos, "Blind Image Deconvolution", *Signal Processing Mag.*, Vol. 13, pp. 43-64, May 1996.
9. D. Kundur and D. Hatzinakos, "Blind Image Deconvolution Revisited", *Signal Processing Mag.*, Vol. 13, pp. 61-63, November 1996.
10. D. Mumford and J. Shah, "Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems", *Communications on Pure and Applied Mathematics*, Vol. 42, pp. 577-684, 1989.
11. R. Nakagaki and A. Katsaggelos, "A VQ-Based Blind Image Restoration Algorithm", *IEEE Trans. Image Processing*, Vol. 12, pp. 1044-1053, 2003.
12. T. Richardson and S. Mitter, "Approximation, Computation and Distortion in the Variational Formulation", in *Geometery-Driven Diffusion in Computer Vision*, B.M. ter Harr Romeny, Ed. Kluwer, Boston, 1994, pp. 169-190.
13. L. Rudin, S. Osher and E. Fatemi, "Non Linear Total Variation Based Noise Removal Algorithms", *Physica D*, Vol. 60, pp. 259-268, 1992.
14. L. Rudin and S. Osher, "Total Variation Based Image Restoration with Free Local Constraints", *Proc. IEEE ICIP*, Vol. 1, pp. 31-35, Austin TX, USA, 1994.
15. C. Samson, L. Blanc-Féraud, G. Aubert and J. Zerubia, "Multiphase Evolution and Variational Image Classification", *Technical Report No. 3662*, INRIA Sophia Antipolis, April 1999.
16. M. Sonka, V. Hlavac and R. Boyle, *Image Processing, Analysis and Machine Vision*, PWS Publishing, 1999.
17. F. Sroubek and J. Flusser, "Multichannel Blind Iterative Image Restoration", *IEEE. Trans. Image Processing*, Vol. 12, pp. 1094-1106, 2003
18. A. Tikhonov and V. Arsenin, "Solutions of Ill-posed Problems", New York, 1977.
19. L.A. Vese and T.F. Chan, "A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model", *International Journal of Computer Vision*, Vol. 50, pp. 271-293, 2002.
20. C. Vogel and M. Oman, "Fast, Robust Total Variation-based Reconstruction of Noisy, Blurred Images", *IEEE Trans. Image Processing*, Vol. 7, pp. 813-824, 1998.
21. Y. You and M. Kaveh, "A Regularization Approach to Joint Blur Identification and Image Restoration", *IEEE Trans. Image Processing*, Vol. 5, pp. 416-428, 1996.

# Towards Intelligent Mission Profiles of Micro Air Vehicles: Multiscale Viterbi Classification

Sinisa Todorovic and Michael C. Nechyba

ECE Department, University of Florida, Gainesville, FL 32611
{sinisha, nechyba}@mil.ufl.edu,
http://www.mil.ufl.edu/mav

**Abstract.** In this paper, we present a vision system for object recognition in aerial images, which enables broader mission profiles for Micro Air Vehicles (MAVs). The most important factors that inform our design choices are: real-time constraints, robustness to video noise, and complexity of object appearances. As such, we first propose the HSI color space and the Complex Wavelet Transform (CWT) as a set of sufficiently discriminating features. For each feature, we then build tree-structured belief networks (TSBNs) as our underlying statistical models of object appearances. To perform object recognition, we develop the novel multiscale Viterbi classification (MSVC) algorithm, as an improvement to multiscale Bayesian classification (MSBC). Next, we show how to globally optimize MSVC with respect to the feature set, using an adaptive feature selection algorithm. Finally, we discuss context-based object recognition, where visual contexts help to disambiguate the identity of an object despite the relative poverty of scene detail in flight images, and obviate the need for an exhaustive search of objects over various scales and locations in the image. Experimental results show that the proposed system achieves smaller classification error and fewer false positives than systems using the MSBC paradigm on challenging real-world test images.

## 1 Introduction

We seek to improve our existing vision system for Micro Air Vehicles (MAVs) [1, 2, 3] to enable more intelligent MAV mission profiles, such as remote traffic surveillance and moving-object tracking. Given many uncertain factors, including variable lighting and weather conditions, changing landscape and scenery, and the time-varying on-board camera pose with respect to the ground, object recognition in aerial images is a challenging problem even for the human eye. Therefore, we resort to a probabilistic formulation of the problem, where careful attention must be paid to selecting sufficiently discriminating features and a sufficiently expressive modeling framework. More importantly, real-time constraints and robustness to video noise are critical factors that inform the design choices for our MAV application.

Having experimented with color and texture features [3], we conclude that both color and texture clues are generally required to accurately discriminate object appearances. As such, we employ both the HSI color space, for color representation, and also the Complex Wavelet Transform (CWT), for multi-scale texture representation. In some cases, where objects exhibit easy-to-classify appearances, the proposed feature set is

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 178–189, 2004.
© Springer-Verlag Berlin Heidelberg 2004

not justifiable in light of real-time processing constraints. Therefore, herein, we propose an algorithm for selecting an optimal feature subspace from the given HSI and CWT feature space that considers both correctness of classification and computational cost.

Given this feature set, we then choose tree-structured belief networks (TSBNs) [4], as underlying statistical models to describe pixel neighborhoods in an image at varying scales. We build TSBNs for both color and wavelet features, using Pearl's message passing scheme [5] and the EM algorithm [6]. Having trained TSBNs, we then proceed with supervised object recognition. In our approach, we exploit the idea of *visual contexts* [7], where initial identification of the overall type of scene facilitates recognition of specific objects/structures within the scene. Objects (e.g., cars, buildings), the locations where objects are detected (e.g., road, meadow), and the category of locations (e.g., sky, ground) form a taxonomic hierarchy. Thus, object recognition in our approach consists of the following steps. First, sky/ground regions in the image are identified. Second, pixels in the ground region[1] are labeled using the learned TSBNs for predefined locations (e.g., road, forest). Finally, pixels of the detected locations of interest are labeled using the learned TSBNs for a set of predefined objects (e.g., car, house).

To reduce classification error (e.g., "blocky segmentation"), which arises from the fixed-tree structure of TSBNs, we develop the novel multiscale Viterbi classification (MSVC) algorithm, an improved version of multiscale Bayesian classification (MSBC) [8, 9]. In the MSBC approach, image labeling is formulated as Bayesian classification at each scale of the tree model, separately; next, transition probabilities between nodes at different scales are learned using the greedy classification-tree algorithm, averaging values over all nodes and over all scales; finally, it is assumed that labels at a "coarse enough" scale of the tree model are statistically independent. On the other hand, in our MSVC formulation, we perform Bayesian classification only at the finest scale, fusing downward the contributions of all the nodes at all scales in the tree; next, transition probabilities between nodes at different scales are learned as histogram distributions that are not averaged over all scales; finally, we assume dependent class labels at the coarsest layer of the tree model, whose distribution we again estimate as a histogram distribution.

## 2 Feature Space

Our feature selection is largely guided by extensive experimentation reported in our prior work [3], where we sought a feature space, which spans both color and texture domains, and whose extraction meets our tight real-time constraints.

We obtained the best classification results when color was represented in the HSI color space. Tests suggested that hue ($H$), intensity ($I$) and saturation ($S$) features were more discriminative, when compared to the inherently highly correlated features of the RGB and other color systems [10]. Also, first-order HSI statistics proved to be sufficient and better than the first and higher-order statistics of other color systems.

For texture-feature extraction, we considered several filtering, model-based and statistical methods. Our conclusion agrees with the comparative study of Randen *et al.* [11],

[1] Recognition of objects in the sky region can be easily incorporated into the algorithm.

which suggests that for problems where many textures with subtle spectral differences occur, as in our case, it is reasonable to assume that spectral decomposition by a filter bank yields consistently superior results over other texture analysis methods. Our experimental results also suggest that it is necessary to analyze both local and regional properties of texture. Most importantly, we concluded that a prospective texture analysis tool must have high directional selectivity. As such, we employ the complex wavelet transform (CWT), due to its inherent representation of texture at different scales, orientations and locations [12]. The CWT's directional selectivity is encoded in six bandpass subimages of complex coefficients at each level, coefficients that are strongly oriented at angles $\pm 15^\circ, \pm 45^\circ, \pm 75^\circ$. Moreover, CWT coefficient magnitudes exhibit the following properties [13, 14]: i) *multi-resolutional* representation, ii) *clustering*, and iii) *persistence* (i.e. propagation of large/small values through scales).

Computing CWT coefficients at all scales and forming a pyramid structure from HSI values, where coarser scales are computed as the mean of the corresponding children, we obtain nine feature trees. These feature structures naturally give rise to TSBN statistical models.

## 3 Tree-Structured Belief Networks

So far, two main types of prior models have been investigated in the statistical image modeling literature – namely, noncausal and causal Markov random fields (MRF). The most commonly used MRF model is the tree-structured belief network (TSBN) [15, 14, 8, 9, 16]. A TSBN is a generative model comprising hidden, $X$, and observable, $Y$, random variables (RVs) organized in a tree structure. The edges between nodes, representing $X$, encode Markovian dependencies across scales, whereas $Y$'s are assumed mutually independent given the corresponding $X$'s, as depicted in Figure 1. Herein, we enable input of observable information, $Y$, also to higher level nodes, preserving the tree dependences among hidden variables. Thus, $Y$ at the lower layers inform the belief network on the statistics of smaller groups of neighboring pixels (at the lowest level, one pixel), whereas $Y$ at higher layers represent the statistics of larger areas in the image. Hence, we enforce the nodes of a tree model to represent image details

(a) (b)

**Fig. 1.** Differences in TSBN models: (a) observable variables at the lowest layer only; (b) our approach: observable variables at all layers. Black nodes denote observable variables and white nodes represent hidden random variables connected in a tree structure.

at various scales.[2] Furthermore, we assume that features are mutually independent, which is reasonable given that wavelets span the feature space using orthogonal basis functions. Thus, our overall statistical model consists of nine mutually independent trees $\mathcal{T}_f$, $f \in \mathcal{F} = \{\pm 15^\circ, \pm 45^\circ, \pm 75^\circ, H, S, I\}$.

In supervised learning problems, as is our case, a hidden RV, $x_i$, assigned to a tree node $i$, $i \in \mathcal{T}_f$, represents a pixel label, $k$, which takes values in a pre-defined set of image classes, $C$. The state of node $i$ is conditioned on the state of its parent $j$ and is specified by conditional probability tables, $P_{ij}^{kl}$, $\forall i, j \in \mathcal{T}_f$, $\forall k, l \in C$. It follows that the joint probability of all hidden RVs, $X = \{x_i\}$, can be expressed as

$$P(X) = \prod_{i,j \in \mathcal{T}_f} \prod_{k,l \in C} P_{ij}^{kl} \ . \tag{1}$$

We assume that the distribution of an observable RV, $y_i$, depends solely on the node state, $x_i$. Consequently, the joint pdf of $Y = \{y_i\}$ is expressed as

$$P(Y|X) = \prod_{i \in \mathcal{T}_f} \prod_{k \in C} p(y_i | x_i = k, \theta_i^k) \ , \tag{2}$$

where $p(y_i|x_i = k, \theta_i^k)$ is modeled as a mixture of $M$ Gaussians,[3] whose parameters are grouped in $\theta_i^k$. In order to avoid the risk of overfitting the model, we assume that the $\theta's$ are equal for all $i$ at the same scale. Therefore, we simplify the notation as $p(y_i|x_i{=}k, \theta_i^k){=}p(y_i|x_i)$. Thus, a TSBN is fully specified by the joint distribution of $X$ and $Y$ given by

$$P(X, Y) = \prod_{i,j \in \mathcal{T}_f} \prod_{k,l \in C} p(y_i|x_i) P_{ij}^{kl} \ . \tag{3}$$

Now, to perform pixel labeling, we face the probabilistic inference problem of computing the conditional probability $P(X|Y)$. In the graphical-models literature, the best-known inference algorithm for TSBNs is Pearl's message passing scheme [5, 18]; similar algorithms have been proposed in the image-processing literature [15, 8, 14]. Essentially, all these algorithms perform belief propagation up and down the tree, where after a number of training cycles, we obtain all the tree parameters necessary to compute $P(X|Y)$. Note that, simultaneously with Pearl's belief propagation, we employ the EM algorithm [6] to learn the parameters of Gaussian-mixture distributions. Since our TSBNs have observable variables at all tree levels, the EM algorithm is naturally performed at all scales. Finally, having trained TSBNs for a set of image classes, we proceed with multiscale image classification.

## 4 Multiscale Viterbi Classification

Image labeling with TSBNs is characterized by "blocky segmentations," due to their fixed-tree structure. Recently, several approaches have been reported to alleviate this

[2] This approach is more usual in the image processing community [8, 14, 9].

[3] For large $M$, a Gaussian-mixture density can approximate any probability density [17].

problem (e.g., [19, 20]), albeit at prohibitively increased computational cost. Given the real-time requirements for our MAV application, these approaches are not realizable, and the TSBN framework remains attractive in light of its linear-time inference algorithms. As such, we resort to our novel multiscale Viterbi classification (MSVC) algorithm to reduce classification error instead.

Denoting all hidden RVs at the leaf level $L$ as $X^L$, classification at the finest scale is performed according to the MAP rule

$$\hat{X}^L = \arg\max_{X^L}\{P(Y|X)P(X)\} = \arg\max_{X^L} g^L. \tag{4}$$

Assuming that the class label, $x_i^l$, of node $i$ at scale $\ell$, completely determines the distribution of $y_i^l$, it follows that:

$$P(Y|X) = \prod_{\ell=L}^{0}\prod_{i\in\ell} p(y_i^\ell|x_i^\ell)\ , \tag{5}$$

where $p(y_i^\ell|x_i^\ell)$ is a mixture of Gaussians, learned using the inference algorithms discussed in Section 3. As is customary for TSBNs, the distribution of $X^\ell$ is completely determined by $X^{\ell-1}$ at the coarser $\ell-1$ scale. However, while, for training, we build TSBNs where each node has only one parent, here, for classification, we introduce a new multiscale structure where we allow nodes to have more than one parent. Thus, in our approach to image classification, we account for horizontal statistical dependencies among nodes at the same level, as depicted in Figure 2. The new multiresolution model accounts for all the nodes in the trained TSBN, except that it no longer forms a tree structure; hence, it becomes necessary to learn new conditional probability tables corresponding to the new edges. In general, the Markov chain rule reads:

$$P(X) = \prod_{\ell=L}^{0}\prod_{i\in\ell} P(x_i^\ell|X^{\ell-1})\ . \tag{6}$$

The conditional probability $P(x_i^\ell|X^{\ell-1})$ in (6), unknown in general, must be estimated using a prohibitive amount of data. To overcome this problem, we consider, for each node $i$, a $3\times 3$ box encompassing parent nodes that neighbor the initial parent $j$ of $i$ in the quad-tree. The statistical dependence of $i$ on other nodes at the next coarser scale,

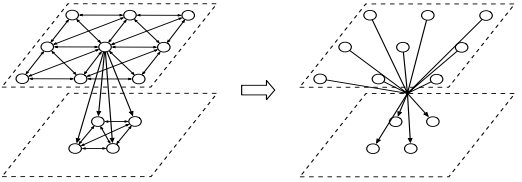


**Fig. 2.** Horizontal dependences among nodes at the same level are modeled by vertical dependences of each node on more than one parent.

in most cases, can be neglected. Thus, we assume that a nine-dimensional vector $v_j^{\ell-1}$, containing nine parents, represents a reliable source of information on the distribution of all class labels $X^{\ell-1}$ for child node $i$ at level $\ell$. Given this assumption, we rewrite expression (6) as

$$P(X) = \prod_{\ell=L}^{0} \prod_{i \in \ell} \prod_{j \in \ell-1} P(x_i^\ell | v_j^\ell) \; . \tag{7}$$

Now, we can express the discriminant function in (4) in a more convenient form as

$$g^L = \prod_{\ell=L}^{0} \prod_{i \in \ell} \prod_{j \in \ell-1} p(y_i^\ell | x_i^\ell) P(x_i^\ell | v_j^{\ell-1}) \; . \tag{8}$$

Assuming that our features $f \in \mathcal{F}$ are mutually independent, the overall maximum discriminant function can therefore be computed as

$$g^L = \prod_{f \in \mathcal{F}} g_f^L \; . \tag{9}$$

The unknown transition probabilities $P(x_i^\ell | v_j^{\ell-1})$ can be learned through vector quantization [21], together with Pearl's message passing scheme. After the prior probabilities of class labels of nodes at all tree levels are learned using Pearl's belief propagation, we proceed with instantiation of random vectors $v_i^\ell$. For each tree level, we obtain a data set of nine-dimensional vectors, which we augment with the class label of the corresponding child node. Finally, we perform vector quantization over the augmented ten-dimensional vectors. The learned histogram distributions represent estimates of the conditional probability tables. Clearly, to estimate the distribution of a ten-dimensional random vector it is necessary to provide a sufficient number of training images, which is readily available from recorded MAV-flight video sequences. Moreover, since we are not constrained by the same real-time constraints during training as during flight, the proposed learning procedure results in very accurate estimates, as is demonstrated in Section 7.

The estimated transition probabilities $P(x_i^\ell | v_j^{\ell-1})$ enable classification from scale to scale in Viterbi fashion. Starting from the highest level downwards, at each scale, we maximize the discriminant function $g^L$ along paths that connect parent and children nodes. From expressions (8) and (9), it follows that image labeling is carried out as

$$\hat{x}_i^L = \arg\max_{x_i^L \in C} \prod_{f \in \mathcal{F}} \prod_{\ell=L}^{0} \prod_{i \in \ell} \prod_{j \in \ell-1} p(y_{i,f}^\ell | x_i^\ell) P(x_i^\ell | \hat{v}_j^{\ell-1}) \; , \tag{10}$$

where $\hat{v}_j^{\ell-1}$ is determined from the previously optimized class labels of the coarser scale $\ell - 1$.

## 5 Adaptive Feature Selection

We have already pointed out that in some cases, where image classes exhibit favorable properties, there is no need to compute expression (10) over all features. Below, we

present our algorithm for adaptive selection of the optimal feature set, $\mathcal{F}_{sel}$, from the initial feature set, $\mathcal{F}$.

1. Form a new empty set $\mathcal{F}_{sel} = \{\emptyset\}$; assign $g_{new} = 1$, $g_{old} = 0$;
2. Compute $\hat{g}_f^L, \forall f \in \mathcal{F}$, given by (8) for $\hat{x}_i^L$ given by (10);
3. Move the best feature $f^*$, for which $\hat{g}_{f^*}^L$ is maximum, from $\mathcal{F}$ to $\mathcal{F}_{sel}$;
4. Assign $g_{new} = \prod_{f \in \mathcal{F}_{sel}} \hat{g}_f^L$;
5. If $(g_{new} < g_{old})$ delete $f^*$ from $\mathcal{F}_{sel}$ and go to step 3;
6. Assign $g_{old} = g_{new}$;
7. If $(\mathcal{F} \neq \{\emptyset\})$ go to step 3;
8. Exit and segment the image using features in $\mathcal{F}_{sel}$.

The discriminant function, $g$, is nonnegative; hence, the above algorithm finds at least one optimal feature. Clearly, the optimization criteria above consider both correctness of classification and computational cost.

## 6 Object Recognition Using Visual Contexts

In our approach to object recognition, we seek to exploit the idea of *visual contexts* [7]. Having previously identified the overall type of scene, we can then proceed to recognize specific objects/structures within the scene. Thus, objects, the locations where objects are detected, and the category of locations form a taxonomic hierarchy. There are several advantages to this type of approach. Contextual information helps disambiguate the identity of objects despite the poverty of scene detail in flight images and quality degradation due to video noise. Furthermore, exploiting visual contexts, we obviate the need for an exhaustive search of objects over various scales and locations in the image.

For each aerial image, we first perform categorization, i.e., sky/ground image segmentation. Then, we proceed with localization, i.e., recognition of global objects and structures (e.g., road, forest, meadow) in the ground region. Finally, in the recognized locations we search for objects of interest (e.g., cars, buildings). To account for different flight scenarios, different sets of image classes can be defined accordingly. Using the prior knowledge of a MAV's whereabouts, we can reduce the number of image classes, and, hence, computational complexity as well as classification error.

At each layer of the contextual taxonomy, downward, we conduct MSVC-based object recognition. Here, we generalize the meaning of image classes to any global-object appearance. Thus, the results from Sections 3 and 4 are readily applicable. In the following example, shown in Figure 3, each element of the set of locations {*road, forest, lawn*} induces subsets of objects, say, {*car, cyclist*} pertaining to *road*. Consequently, when performing MSVC, we consider only a small finite number of image classes, which improves recognition results. Thus, in spite of video noise and poverty of image detail, the object in Figure 3, being tested against only two possibilities, is correctly recognized as a *car*.

## 7 Results

In this section, we demonstrate the performance of the proposed vision system for real-time object recognition in supervised-learning settings. We carried out several sets of

(a) (b) (c) (d)

**Fig. 3.** The hierarchy of visual contexts conditions gradual image interpretation: (a) a $128 \times 128$ flight image; (b) categorization: sky/ground classification; (c) localization: road recognition; (d) car recognition.

experiments which we report below. For space reasons, we discuss only our results for car and cyclist recognition in flight video.

For training TSBNs, we selected 200 flight images for the car and cyclist classes. We carefully chose the training sets to account for the enormous variability in car and cyclist appearances, as illustrated in Figure 4 (top row). After experimenting with different image resolutions, we found that reliable labeling was achievable for resolutions as coarse as $64 \times 64$ pixels. At this resolution, all the steps in object recognition (i.e., sky/ground classification, road localization and car/cyclist recognition), when the feature set comprises all nine features, takes approximately 0.1s on an Athlon 2.4GHz PC. For the same set-up, but for only one optimal feature, recognition time is less than 0.07s,[4] which is quite sufficient for the purposes of moving-car or moving-bicycle tracking. Moreover, for a sequence of video images, the categorization and localization steps could be performed only for images that occur at specified time intervals, although, in our implementation, we process every image in a video sequence for increased noise robustness.

After training our car and bicycle statistical models, we tested MSVC performance on a set of 100 flight images. To support our claim that MSVC outperforms MSBC, we carried out a comparative study of the two approaches on the same dataset. For validation accuracy, we separated the test images into two categories. The first category consists of 50 test images with easy-to-classify car/cyclist appearances as illustrated in Figure 4a and Figure 4b. The second category includes another 50 images, where multiple hindering factors (e.g. video noise and/or landscape and lighting variability, as depicted in Figure 4c and Figure 4d) conditioned poor classification. Ground truth was established through hand-labeling pixels belonging to objects for each test image. Then, we ran the MSVC and MSBC algorithms, accounting for the image-dependent optimal subset of features. Comparing the classification results with ground truth, we computed the percentage of erroneously classified pixels for the MSVC and MSBC algorithms. The results are summarized in Table 1, where we do not report the error of complete

[4] Note that even if only one set of wavelet coefficients is optimal, it is necessary to compute all other sets of wavelets in order to compute the optimal one at all scales. Thus, in this case, time savings are achieved only due to the reduced number of features for which MSVC is performed.

(a) (b) (c) (d)

**Fig. 4.** Recognition of road objects: (top) Aerial flight images; (middle) localization: road recognition; (bottom) object recognition. MSVC was performed for the following optimized sets of features: (a) $\mathcal{F}_{sel} = \{H, I, -45^\circ\}$, (b) $\mathcal{F}_{sel} = \{H, \pm 75^\circ\}$, (c) $\mathcal{F}_{sel} = \{\pm 15^\circ, \pm 45^\circ\}$, (d) $\mathcal{F}_{sel} = \{H, \pm 45^\circ\}$.

misses (CM) (i.e., the error when an object was not detected at all) and the error of swapped identities (SI) (i.e., the error when an object was detected but misinterpreted). Also, in Table 2, we report the recognition results for 86 and 78 car/cyclist objects in the first and second categories of images, respectively. In Figure 5, we illustrate better MSVC performance over MSBC for a sample first-category image.

**Table 1.** Percentage of misclassified pixels by MSVC and MSBC

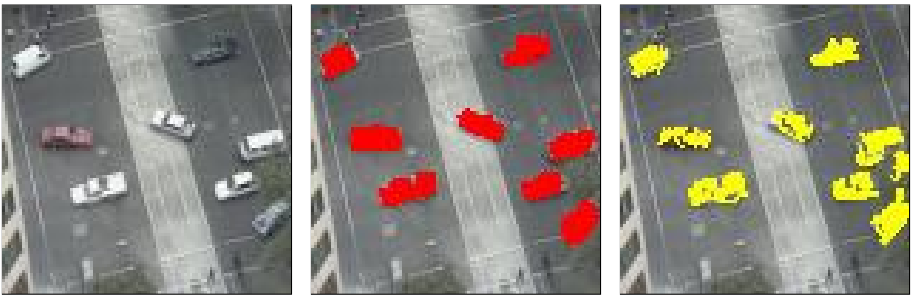
| | I category images | II category images |
|---|---|---|
| MSVC | 4% | 10% |
| MSBC | 9% | 17% |

Finally, we illustrate the validity of our adaptive feature selection algorithm. In Figure 6, we present MSVC results for different sets of features. Our adaptive feature selection algorithm, for the given image, found $\mathcal{F}_{sel} = \{H, -45^\circ, \pm 75^\circ\}$ to be the opti-

mal feature subset. To validate the performance of the selection algorithm, we segmented the same image using all possible subsets of the feature set $\mathcal{F}$. For space reasons, we illustrate only some of these classification results. Obviously, from Figure 6, the selected optimal features yield the best image labeling. Moreover, note that when all the features were used in classification, we actually obtained worse results. In Table 3, we present the percentage of erroneously classified pixels by MSVC using different subsets of features for our two categories of 100 test images. As before, we do not report the error of complete misses. Clearly, the best classification results were obtained for the optimal set of features.

**Table 2.** Correct recognition (CR), complete miss (CM), and swapped identity (SI)

| | I category images (86 objects) | | | II category images (78 objects) | | |
|---|---|---|---|---|---|---|
| | CR | CM | SI | CR | CM | SI |
| MSVC | 81 | 1 | 4 | 69 | 5 | 4 |
| MSBC | 78 | 2 | 6 | 64 | 9 | 5 |

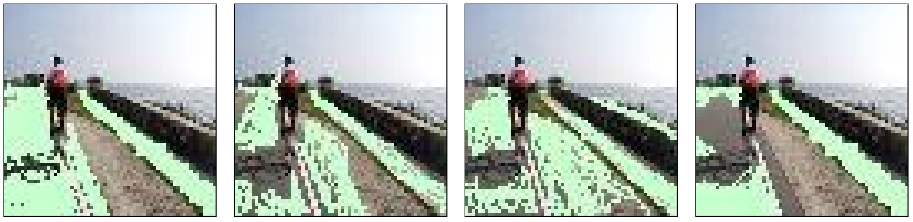


(a) (b) (c)

**Fig. 5.** Better performance of MSVC vs. MSBC for the optimal feature set $\mathcal{F}_{sel} = \{H, I, \pm 15^\circ, \pm 75^\circ\}$: (a) a first-category image; (b) MSVC; (c) MSBC.

## 8 Conclusion

Modeling complex classes in natural-scene images requires an elaborate consideration of class properties. The most important factors that informed our design choices for a MAV vision system are: (1) real-time constraints, (2) robustness to video noise, and (3) complexity of various object appearances in flight images. In this paper, we first presented our choice of features: the HSI color space, and the CWT. Then, we introduced



(a) (b) (c) (d)

**Fig. 6.** Validation of the feature selection algorithm for road recognition: (a) MSVC for the optimized $\mathcal{F}_{sel} = \{H, -45^\circ, \pm 75^\circ\}$; (b) MSVC for all nine features in $\mathcal{F}$; (c) MSVC for subset $\mathcal{F}_1 = \{H, S, I\}$; (d) MSVC for subset $\mathcal{F}_2 = \{\pm 15^\circ, \pm 45^\circ \pm 75^\circ\}$.

**Table 3.** Percentage of misclassified pixels by MSVC

| | I category | II category |
|---|---|---|
| $\mathcal{F}_{sel}$ | 4% | 10% |
| $\mathcal{F} = \{H, S, I, \pm 15^\circ, \pm 45^\circ \pm 75^\circ\}$ | 13% | 17% |
| $\mathcal{F}_1 = \{H, S, I\}$ | 16% | 19% |
| $\mathcal{F}_2 = \{\pm 15^\circ, \pm 45^\circ \pm 75^\circ\}$ | 14% | 17% |

the TSBN model and the training steps for learning its parameters. Further, we described how the learned parameters could be used for computing the likelihoods of all nodes at all TSBN scales. Next, we proposed and demonstrated multiscale Viterbi classification (MSVC), as an improvement to multiscale Bayesian classification. We showed how to globally optimize MSVC with respect to the feature set through an adaptive feature selection algorithm. By determining an optimal feature subset, we successfully reduced the dimensionality of the feature space, and, thus, not only approached the real-time requirements for applications operating on real-time video streams, but also improved overall classification performance. Finally, we discussed object recognition based on visual contexts, where contextual information helps disambiguate the identity of objects despite a poverty of scene detail and obviates the need for an exhaustive search of objects over various scales and locations in the image. We organized test images into two categories of difficulty and obtained excellent classification results, especially for complex-scene/noisy images, thus validating the proposed approach.

## References

1. Ettinger, S.M., Nechyba, M.C., Ifju, P.G., Waszak, M.: Vision-guided flight stability and control for Micro Air Vehicles. In: Proc. IEEE Int'l Conf. Intelligent Robots and Systems (IROS), Laussane, Switzerland (2002)
2. Ettinger, S.M., Nechyba, M.C., Ifju, P.G., Waszak, M.: Vision-guided flight stability and control for Micro Air Vehicles. Advanced Robotics **17** (2003)
3. Todorovic, S., Nechyba, M.C., Ifju, P.: Sky/ground modeling for autonomous MAVs. In: Proc. IEEE Int'l Conf. Robotics and Automation (ICRA), Taipei, Taiwan (2003)

4. Cowell, R.G., Dawid, A.P., Lauritzen, S.L., Spiegelhalter, D.J.: Probabilistic Networks and Expert Systems. Springer-Verlag, New York (1999)
5. Pearl, J.: Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference. Morgan Kaufamnn, San Mateo (1988)
6. McLachlan, G.J., Thriyambakam, K.T.: The EM algorithm and extensions. John Wiley & Sons (1996)
7. Torralba, A., Murphy, K.P., Freeman, W.T., Rubin, M.A.: Context-based vision system for place and object recognition. In: Proc. Int'l Conf. Computer Vision (ICCV), Nice, France (2003)
8. Cheng, H., Bouman, C.A.: Multiscale bayesian segmentation using a trainable context model. IEEE Trans. Image Processing **10** (2001)
9. Choi, H., Baraniuk, R.G.: Multiscale image segmentation using wavelet-domain Hidden Markov Models. IEEE Trans. Image Processing **10** (2001)
10. Cheng, H.D., Jiang, X.H., Sun, Y., Jingli, W.: Color image segmentation: advances and prospects. Pattern Recognition **34** (2001)
11. Randen, T., Husoy, H.: Filtering for texture classification: A comparative study. IEEE Trans. Pattern Analysis Machine Intelligence **21** (1999)
12. Kingsbury, N.: Image processing with complex wavelets. Phil. Trans. Royal Soc. London **357** (1999)
13. Mallat, S.: A Wavelet Tour of Signal Processing. 2nd edn. Academic Press (2001)
14. Crouse, M.S., Nowak, R.D., Baraniuk, R.G.: Wavelet-based statistical signal processing using Hidden Markov Models. IEEE Trans. Signal Processing **46** (1998)
15. Bouman, C.A., Shapiro, M.: A multiscale random field model for Bayesian image segmentation. IEEE Trans. Image Processing **3** (1994)
16. Feng, X., Williams, C.K.I., Felderhof, S.N.: Combining belief networks and neural networks for scene segmentation. IEEE Trans. Pattern Analysis Machine Intelligence **24** (2002)
17. Aitkin, M., Rubin, D.B.: Estimation and hypothesis testing in finite mixture models. J. Royal Stat. Soc. **B-47** (1985)
18. Frey, B.J.: Graphical Models for Machine Learning and Digital Communication. The MIT Press, Cambridge, MA (1998)
19. Storkey, A.J., Williams, C.K.I.: Image modeling with position-encoding dynamic trees. IEEE Trans. Pattern Analysis Machine Intelligence **25** (2003)
20. Irving, W.W., Fieguth, P.W., Willsky, A.S.: An overlapping tree approach to multiscale stochastic modeling and estimation. IEEE Trans. Image Processing **6** (1997)
21. Linde, Y., Buzo, A., Gray, R.M.: An algorithm for vector quantizer design. IEEE Trans. on Communications **COM-28** (1980)

# Stitching and Reconstruction of Linear-Pushbroom Panoramic Images for Planar Scenes

Chu-Song Chen[1], Yu-Ting Chen[2,1], and Fay Huang[1]

[1] Institute of Information Science, Academia Sinica,
Nankang, Taipei, Taiwan, R. O. C.,
song@iis.sinica.edu.tw
[2] Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan, R.O.C.

**Abstract.** This paper proposes a method to integrate multiple linear-pushbroom panoramic images. The integration can be performed in real time. The technique is feasible on planar scene such as large-scale paintings or aerial/satellite images that are considered to be planar. The image integration consists of two steps: stitching and Euclidean reconstruction. For the image stitching, a minimum of five pairs of non-collinear image corresponding points are required in general cases. In some special configurations when there is column-to-column image correspondence between two panoramas, the number of image corresponding points required can be reduced to three. As for the Euclidean reconstruction, five pairs of non-collinear image corresponding points on the image boundaries are sufficient.

## 1 Introduction

The image mosaicing techniques have been applied in photogrammetry back in 80's for constructing large aerial and satellite photographs [16]. However, only until 90's, intensive researches on automatic construction of panoramic image mosaics were carried out in the fields of computer vision and computer graphics [20]. Two main features of the image mosaicing concept are the abilities to increase the resolution and to enlarge the field of view of a camera.

In computer vision, panoramic image mosaics often serve as representations of visual scene for a wide diversity of applications [14, 21, 6, 12, 8]. In the case when multiple panoramic images are provided, the depth information or other geometric properties of the 3D scene can be recovered [11, 18, 22]. In computer graphics, panoramic image mosaics play an important role in the technique of image-based rendering [2, 15, 4, 13, 10, 19]. The key idea of this technique is to rapidly generate novel views from a set of existing images. Panoramic images are also used widely in virtual reality systems to provide an immersive and photorealistic environment [1].

The traditional way of constructing a panoramic image mosaic is to align a set of matrix images of a common view by performing image transformations. When

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 190–201, 2004.
© Springer-Verlag Berlin Heidelberg 2004

images are acquired by unknown camera poses, there is a need to solve camera calibration problem before the image transformations may take place. Up until more recently, the line-camera concept for creating panoramic image mosaics were introduced [9, 5, 17], in which a sequence of slit (or line) images are used as the base elements instead of the matrix images for composing a panoramic image. Such a panorama is generated by joining together a sequence of line images side by side, and is called line-based panoramic images. The main advantage of using line images is to ease or even to avoid the camera extrinsic calibration problem so that panoramic mosaics can be generated simultaneously during the image acquisition process. One major trade-off of the line-based panoramic images is that the vertical image field of view is constrained by the resolution of the line image. There is lack of research on stitching two line-based panoramic images vertically to increasing the panorama's field of view.

Linear-pushbroom camera model was first introduced by R. I. Hartley in 1997 [5], which belongs in the line-based panorama category. The main characteristic of linear-pushbroom panoramic images is that the line-camera moves along a straight line during image acquisition. We investigated the possibility of integrating two such panoramic images under some additional geometric constrains. It is found that two linear-pushbroom panoramas are geometrically related by an affine transformation if they capture a common planar scene. In this paper, the integration of two linear-pushbroom panoramic images of planar scene is established for the first time. We conclude that only few image corresponding points are needed to perform the integration. This integration technique can be used for digitizing the large-scale 2D artworks in the museums or documenting the huge historical paintings on the wall. It can also be used on aerial or satellite images that are considered to be planar.

The paper is organized as follows: linear-pushbroom camera model is summarized in Section 2, in which the projection matrix and the LP-fundamental matrix of this camera model are recapped. The image integration method is reported in Section 3, in which the image transformation equations for image stitching and Euclidean reconstruction are elaborated respectively. Section 4 illustrates the integration result of a large-scale famous Chinese painting. Finally, conclusions are drawn in Section 5.

## 2 Review of Linear-Pushbroom Camera Model

A linear-pushbroom camera can be considered as a perspective line-camera[1] moving in a linear orbit with a constant velocity and a fixed orientation. As the line-camera moves, the view plane[2] sweeps out a region of space and 1-D images are captured. Finally, the whole 1-D images constitute a 2-D image which lies on a plane called the image plane in 3-D space.

[1] An optical system projecting an image onto a 1-D array sensor, typically a CCD array, is called a line-camera.

[2] The plane defined by the optical center and the sensor array is called a view plane.

An arbitrary point $\mathbf{x} = (x, y, z)^T$ in space is imaged and represented by two coordinates $u$ and $v$. It has been shown in [5] that the linear-pushbroom camera model can be conducted as follows:

$$(u, wv, w)^T = \mathbf{M}(x, y, z, 1)^T \tag{1}$$

where $w$ is a scale factor and $\mathbf{M}$ is a 3×4 projection matrix. The linear-pushbroom camera model, $(u, wv, w)^T = \mathbf{M}(x, y, z, 1)^T$, should be compared with the basic pin-hole camera model. An obvious difference is that the matrix of the pin-hold camera model is homogeneous; however, the linear-pushbroom camera matrix is not. That is, by multiplying linear-pushbroom camera matrix $\mathbf{M}$ with an arbitrary factor $k$, the $v$ coordinate is unchanged while the $u$ coordinate is scaled by $k$.

Consider a point $\mathbf{x} = (x, y, z)^T$ in space viewed by two linear-pushbroom cameras with projection matrices $\mathbf{M}$ and $\mathbf{M}'$. Let $\mathbf{u} = (u, v)^T$ and $\mathbf{u}' = (u', v')^T$ be the mappings of point $\mathbf{x}$ on these two panoramas respectively. A cubic equation $p(u, v, u', v') = 0$ called *fundamental polynomial* corresponding to these two cameras is introduced in [5], where the coefficients of $p$ are determined by the entries of $\mathbf{M}$ and $\mathbf{M}'$. It concludes in the paper that there exists a 4×4 matrix $F$ such that the equation $p(u, v, u', v') = 0$ may be rewritten as follows:

$$(u', u'v', v', 1)F_{4\times4}(u, uv, v, 1)^T = 0.$$

The matrix $F$ is called the *LP-fundamental matrix* corresponding to the linear-pushbroom camera pair $\{\mathbf{M}, \mathbf{M}'\}$. The matrix expresses the relationships between corresponding curves in these two linear-pushbroom panoramic images.

## 3 Integration of Linear-Pushbroom Panoramic Images

Consider two linear-pushbroom panoramic images (or LP-mosaic images) targeting at a planar scene (such as a large painting on the wall). In this section, we propose an image integration method to stitch these two panoramic images by image correspondence information. A key point to achieve this purpose is that, for an arbitrary point on the first image, can its corresponding point in the second image be determined and vice versa? In general, only curve-to-curve relationships can be established for two LP-mosaic images according to the theory of LP-fundamental matrix. Hence, the correspondence is ambiguous (to a curve) for a point specified in the first image and vice versa.

For perspective case, the point-to-point relationship can be established by imposing some scene constraints, such as co-planarity. The so-called planar homography [3] can be determined by four given pairs of image correspondences, and a complete point-to-point relationship can be exactly established from the planar homography thus determined. Planar homography has been widely adopted for applications such as image mosaicing [1, 21] and panorama construction [20].

As to the linear-pushbroom case, what we are interested is whether the point-to-point relationships can be determined as well when the scene is planar? If the

answer is yes, how many image correspondence pairs are required? These issues will be addressed in the following.

### 3.1 Image Stitching

Let $\mathbf{x}_i = (x_i, y_i, z_i)^T$ denote points in space that lie on a plane with planar equation $E : ax_i + by_i + cz_i + d = 0$ and are viewed by two linear-pushbroom cameras. Let $\mathbf{u}_i = (u_i, v_i)^T$ and $\mathbf{u}'_i = (u'_i, v'_i)^T$ be the mapping of point $\mathbf{x}_i$ on the source and the destination LP-mosaic images respectively. We intend to find transformation equations, which transform all the image points of the source panorama to the destination panorama, based on a set of corresponding points $\mathbf{u}_i$ and $\mathbf{u}'_i$.

According to linear-pushbroom camera model discussed in the last section (equation 1), we have

$$\begin{array}{l}(u_i, w_i v_i, w_i)^T = \mathbf{M}(x_i, y_i, z_i, 1)^T \\ (u'_i, w'_i v'_i, w'_i)^T = \mathbf{M}'(x_i, y_i, z_i, 1)^T\end{array} \tag{2}$$

where $\mathbf{M}$ and $\mathbf{M}'$ are $3 \times 4$ projection matrix associated to the source and the destination panoramic images, respectively. Let $m_{jk}$ and $m'_{jk}$, where $1 \leq j \leq 3$ and $1 \leq k \leq 4$ denote the elements of $\mathbf{M}$ and $\mathbf{M}'$ respectively. Equation 2 plus the planar equation $E : ax_i + by_i + cz_i + d = 0$ can be rearranged into the following seven equations:

$$\begin{cases} u_i = m_{11}x_i + m_{12}y_i + m_{13}z_i + m_{14} \ldots \text{(i)} \\ w_i v_i = m_{21}x_i + m_{22}y_i + m_{23}z_i + m_{24} \ldots \text{(ii)} \\ w_i = m_{31}x_i + m_{32}y_i + m_{33}z_i + m_{34} \ldots \text{(iii)} \\ u'_i = m'_{11}x_i + m'_{12}y_i + m'_{13}z_i + m'_{14} \ldots \text{(iv)} \\ w'_i v'_i = m'_{21}x_i + m'_{22}y_i + m'_{23}z_i + m'_{24} \ldots \text{(v)} \\ w'_i = m'_{31}x_i + m'_{32}y_i + m'_{33}z_i + m'_{34} \ldots \text{(vi)} \\ ax_i + by_i + cz_i + d = 0 \ldots \text{(vii)} \end{cases} \tag{3}$$

Because $w_i$ and $w'_i$ are not necessary to be the same for each corresponding pair $\mathbf{u}_i$ and $\mathbf{u}'_i$, we may deal with these two variables separately. First, for a given $\mathbf{u}_i$, we use equations (i), (ii), (iii), (iv), and (vii) in equation 3 to find its corresponding $u'_i$ to avoid the influence of $w'_i$, and the following equation holds:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} - u_i & 0 \\ m_{21} & m_{22} & m_{23} & m_{24} & v_i \\ m_{31} & m_{32} & m_{33} & m_{34} & 1 \\ m'_{11} & m'_{12} & m'_{13} & m'_{14} - u'_i & 0 \\ a & b & c & d & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \\ -w_i \end{bmatrix} = 0, \tag{4}$$

where the left $5 \times 5$ matrix is denoted as $W_1$. This is a set of five homogeneous equations with five unknowns. Because one of the five unknowns is 1, this means that this equation has a non-zero solution. Note that only $\det(W_1) = 0$ can allow equation 4 to have a non-zero solution. Because the determinant of $W_1$ consists

of terms in $u_i$, $v_i$, $u'_i$, $u_iv_i$, and $u'_iv_i$, the following equation with six coefficients, $a_0 \sim a_5$, exists:

$$a_0 + a_1u_i + a_2v_i + a_3u'_i + a_4u_iv_i + a_5u'_iv_i = 0. \tag{5}$$

Similarly, for a given $\mathbf{u}'_i$, we use equations (i), (iv), (v), (vi), and (vii) in equation 3 to find its corresponding $u_i$ to avoid the influence of $w_i$. We again obtain a set of five homogeneous equations with five unknowns. By the same argument as above, we may conclude that the following equation with six coefficients, $b_0 \sim b_5$, exists:

$$b_0 + b_1u_i + b_2u'_i + b_3v'_i + b_4u_iv'_i + b_5u'_iv'_i = 0.$$

Suppose $a_i$ and $b_i$ are known, given a point $\mathbf{u}_i = (u_i, v_i)^T$, its corresponding point $\mathbf{u}'_i = (u'_i, v'_i)^T$ can be calculated by the following equations:

$$\begin{cases} u'_i = \frac{-(a_0+a_1u_i+a_2v_i+a_4u_iv_i)}{a_3+a_5v_i} \\ v'_i = \frac{-(b_0+b_1u_i+b_2u'_i)}{b_3+b_4u_i+b_5u'_i}. \end{cases} \tag{6}$$

These equations can be applied to transform all the image points of one panorama to the other panorama. After the transformation, we obtain an panoramic image. Therefore, the problem left to be solved is determining the values of the twelve coefficients, $a_0 \sim a_5$ and $b_0 \sim b_5$, by image corresponding points provided.

Given $n$ pairs of corresponding points, namely $\{u_i, v_i, u'_i.v'_i\}$, $i \in [1..n]$, where not all image points $(u_i, v_i)^T$ lie on the same image row or image column. Equation 5 can be restated as follows:

$$a_0\mathbf{1} + a_1\mathbf{U} + a_2\mathbf{V} + a_3\mathbf{U}' + a_4\mathbf{W} + a_5\mathbf{W}' = \mathbf{0}, \tag{7}$$

where $\mathbf{1}$, $\mathbf{U}$, $\mathbf{V}$, $\mathbf{U}'$, $\mathbf{W}$, $\mathbf{W}'$ and $\mathbf{0}$ are all $n$-vectors. Note that vector $\mathbf{1}$ has all its elements equal to one and vector $\mathbf{0}$ has all elements equal to zero. In order to have a non-trivial solution of $a_0 \sim a_5$, the rank of matrix $[\mathbf{1}, \mathbf{U}, \mathbf{V}, \mathbf{U}', \mathbf{W}, \mathbf{W}']$ must equal to five. Moreover, if $\{a_0, a_1, a_2, a_3, a_4, a_5\}$ is a solution of equation 7, then $\{ka_0, ka_1, ka_2, ka_3, ka_4, ka_5\}$ will also be solutions for all $k \in \mathbb{R}$. However, all solutions lead to the same value of $u'_i$ as shown in equation 6. Hence, we aim to find any set of $a_0 \sim a_5$ that satisfies equation 7.

Since there are six unknowns and the six-dimensional solution vector is up to a common scale factor, equation 7 can be solved with at least five pairs of image correspondences. In our work, we assume $a_0^2 + a_1^2 + \ldots + a_5^2 = 1$. When $n \geq 5$, a least-squared-error solution can be obtained by solving the eigenvalue problem in association with the scatter matrix of the linear equation system. Similar arguments also apply to $b_0 \sim b_5$.

Singular case occurs when the rank of matrix $[\mathbf{U}, \mathbf{V}, \mathbf{U}', \mathbf{W}, \mathbf{W}']$ is less than five. A common situation which causes the singular case is when vectors $\mathbf{U}$ and $\mathbf{U}'$ are linearly dependent, that is when we have $\mathbf{U} = A\mathbf{U}' + B$ for some $A, B \in \mathbb{R}$. This situation happens when the two line sensors used for grabbing the two LP-mosaic images are parallel to each other. (Note: we explain this situation in Appendix.) We despite the case when there are only few image

corresponding points provided, so two or more vectors of $\mathbf{U}$, $\mathbf{V}$, $\mathbf{U}'$, $\mathbf{W}$, and $\mathbf{W}'$ happen to be linear dependent because of poor sampling.

When vectors $\mathbf{U}$ and $\mathbf{U}'$ are linearly dependent, instead of using equation 6, we derive another set of transformation equations to transform the image. First, since we know $\mathbf{U} = A\mathbf{U}' + B$, the values of $A$ and $B$ can be obtained straightforwardly by solving a system of linear equations with at least two pairs of image correspondences. Secondly, by substituting equations (i) and (iv) in equation 3 into $u_i = Au'_i + B$, we get

$$\begin{cases} (m_{11} - Am'_{11})x_i + (m_{12} - Am'_{12})y_i + (m_{13} - Am'_{13})z_i + (m_{14} - Am'_{14} - B) = 0 \\ m_{21}x_i + m_{22}y_i + m_{23}z_i - v_i w_i + m_{24} = 0 \\ m_{31}x_i + m_{32}y_i + m_{33}z_i - w_i + m_{34} = 0 \\ m'_{21}x_i + m'_{22}y_i + m'_{23}z_i - v'_i w'_i + m'_{24} = 0 \\ m'_{31}x_i + m'_{32}y_i + m'_{33}z_i - w'_i + m'_{34} = 0 \\ ax_i + by_i + cz_i + d = 0. \end{cases}$$

It implies

$$\begin{bmatrix} m_{11} - Am'_{11} & m_{12} - Am'_{12} & m_{13} - Am'_{13} & m_{14} - Am'_{14} - B & 0 & 0 \\ m_{21} & m_{22} & m_{23} & m_{24} & -v_i & 0 \\ m_{31} & m_{32} & m_{33} & m_{34} & -1 & 0 \\ m'_{21} & m'_{22} & m'_{23} & m'_{24} & 0 & -v'_i \\ m'_{31} & m'_{32} & m'_{33} & m'_{34} & 0 & -1 \\ a & b & c & d & 0 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \\ w_i \\ w'_i \end{bmatrix} = 0,$$

where the left $6 \times 6$ matrix is denoted as $W_2$. It has non-trivial solution if $\det(W_2) = 0$, which means there exist coefficients $c_0 \sim c_3$ such that the following equation holds:

$$c_0 v_i + c_1 v'_i + c_2 v_i v'_i + c_3 = 0.$$

Given at least three pairs of image corresponding points, we are able to determine solutions of $c_0 \sim c_3$. The resulting transformation equations in this case are as follows:

$$\begin{cases} u'_i = \frac{u_i - B}{A} \\ v'_i = \frac{-c_0 v_i - c_3}{c_1 + c_2 v_i}. \end{cases}$$

### 3.2 Euclidean Reconstruction

Through the stitching method introduced above, two LP-mosaic images can be integrated into a single one. In our work, one of the LP-mosaic images is selected as the reference image, and the other image is transformed (or wrapped) so that the two images can be registered. By doing so, the integrated panoramic image can be treated as being obtained by enlarging the viewing range of the line sensor used for grabbing the reference image and then employing this range-enlarged sensor to scan the same planar scene. However, such a scanning process cannot preserve some desired properties such as right angle and parallelism.

From the reconstruction point of view, only an affine reconstruction can be obtained by using only image correspondence information as introduced in [5]. Without imposing other constraints on the scene or the camera, Euclidean reconstruction is impossible. Availability of constraints is application dependent. In

the following, we focus on the application of reconstructing a large-scale painting. The priori knowledge that the painting is of rectangular shape provides scene constraints for Euclidean reconstruction.

To upgrade the original reconstruction to a Euclidean one, let us image a virtual line sensor that scans the painting in the way that this sensor is aligned with the painting plane and is moved on this plane (that is, the painting is scanned by a 'virtual' Xerox machine). Furthermore, we assume that the line sensor is placed parallel to one of the four borderlines of the paintings, and the moving path is perpendicular to the line sensor. Then, the painting scanned with this virtual camera shall be of rectangular shape as well, and we call this rectangular frame the destination image. We aim to reconstruct the rectangular frame by performing some transformation from the integrated panoramic image to the destination image. However, there is no image pixel information for the virtual rectangular frame. Thus, the only correspondence knowledge can be applied are the boundaries of the frame and the image. By assuming that the ratio of the width to the height of the rectangular frame is known, we will show that a Euclidean reconstruction can be achieved by employing the four pairs of borderline-to-borderline correspondences of the painting.

Let $(u_i, v_i)$ be the image coordinates of the source panoramic image (the integrated mosaic), and $(u'_i, v'_i)$ be the image coordinates of the destination rectangular frame. Based on the linear-pushbroom camera model as discussed in Section 2, we have $(u'_i, w'_i v'_i, w'_i)^T = \mathbf{M'}(x_i, y_i, z_i, 1)^T$. Since the destination image is obtained by a 'virtual' Xerox machine, $w'_i$ becomes constant for all $i$ and the value $w'_i$ can be absorbed by the second and the third row of matrix $\mathbf{M'}$. Hence, we have

$$\begin{aligned}(u_i, w_i v_i, w_i)^T &= \mathbf{M}(x_i, y_i, z_i, 1)^T \\ (u'_i, v'_i, 1)^T &= \mathbf{M'}(x_i, y_i, z_i, 1)^T\end{aligned}$$

and these two equations can be expanded as follows:

$$\begin{cases} u_i = m_{11}x_i + m_{12}y_i + m_{13}z_i + m_{14} \ldots \text{(i)} \\ w_i v_i = m_{21}x_i + m_{22}y_i + m_{23}z_i + m_{24} \ldots \text{(ii)} \\ w_i = m_{31}x_i + m_{32}y_i + m_{33}z_i + m_{34} \ldots \text{(iii)} \\ u'_i = m'_{11}x_i + m'_{12}y_i + m'_{13}z_i + m'_{14} \ldots \text{(iv)} \\ v'_i = m'_{21}x_i + m'_{22}y_i + m'_{23}z_i + m'_{24} \ldots \text{(v)} \\ 1 = m'_{31}x_i + m'_{32}y_i + m'_{33}z_i + m'_{34} \ldots \text{(vi)} \end{cases} \tag{8}$$

From (i), (iv), (v) and (vi) in equation 8, we obtain the following:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} - u_i \\ m'_{11} & m'_{12} & m'_{13} & m'_{14} - u'_i \\ m'_{21} & m'_{22} & m'_{23} & m'_{24} - v'_i \\ m'_{31} & m'_{32} & m'_{33} & m'_{34} - 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} = 0,$$

The determinate of the left $4 \times 4$ matrix must equal to zero, hence there exist coefficients $d_0 \sim d_3$, such that the following equation holds:

$$d_0 u_i + d_1 u'_i + d_2 v'_i + d_3 = 0. \tag{9}$$

Moreover, from (ii), (iii), (iv), (v) and (vi) in equation 8, we have

$$\begin{bmatrix} m_{21} & m_{22} & m_{23} & m_{24} & -v_i \\ m_{31} & m_{32} & m_{33} & m_{34} & -1 \\ m'_{11} & m'_{12} & m'_{13} & m'_{14} - u'_i & 0 \\ m'_{21} & m'_{22} & m'_{23} & m'_{24} - v'_i & 0 \\ m'_{31} & m'_{32} & m'_{33} & m'_{34} - 1 & 0 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \\ w_i \end{bmatrix} = 0,$$

and by the same reason as above, we obtain the following equation:

$$e_0 v_i + e_1 u'_i + e_2 v'_i + e_3 v_i u'_i + e_4 v_i v'_i + e_5 = 0. \tag{10}$$

Let the four corner points of the virtual rectangular frame be $(0, 0)$, $(W, 0)$, $(0, H)$, and $(W, H)$, respectively, where $W$ and $H$ are the width and height of the frame. Use these corner points as four inputs $(u'_i, v'_i)$ together with the corresponding corners of the integrated panoramic image $(u_i, v_i)$, we are able to determine the values of $d_0 \sim d_3$ in equation 9.

Consider a boundary point $(u_i, v_i)^T$ lying on one of the border lines of the painting in the integrated panoramic image. One of its corresponding values $u'_i$ and $v'_i$ is known, (which is equal to either $W$ or $H$), because the painting's borderline-to-borderline correspondences are available. The unknown one can be obtained by equation 9 based on the determined values of $d_0 \sim d_3$. Thus, by given at least five image correspondences on the boundaries, we are able to determine the values of $e_0 \sim e_5$ in equation 10. Once all the values of the coefficients are known, the transformation equations can be derived. This transformation enables us to refine the integrated panoramic image to a Euclidean reconstruction.

## 4 Experimental Results

We conducted a synthetic experiment to demonstrate how the image correspondence error (i.e. the input noise) affects the image stitching result. The experiment was designed as follows. There are 250 coplanar points randomly distributed in a bounded space. For each trial, two linear-pushbroom panoramic images with image resolutions of $170 \times 550$ are captured by two virtual line-cameras, whose intrinsic parameters are identical and set to be constant during the image acquisition. The starting positions and the moving velocities of these two cameras vary in each trial. The values of the position and the velocity vectors are randomly chosen within practical ranges. The image correspondence error is introduced by corrupting the ideal image projections by some random noise up to two image pixels. The image stitching error is measured as the average square-norm distance of all pairs of image corresponding points after merging. The average stitching error of 1000 trials is calculated for each noise scale.

Figure 1 is an illustration of our synthetic experiment. The image corresponding points with labels are shown in two linear-pushbroom panoramic images as well as in resulting image after stitching process. Table 1 summarizes our experiment results, which suggests that the stitching error increases linearly as the input noise increases linearly from zero to two pixels.

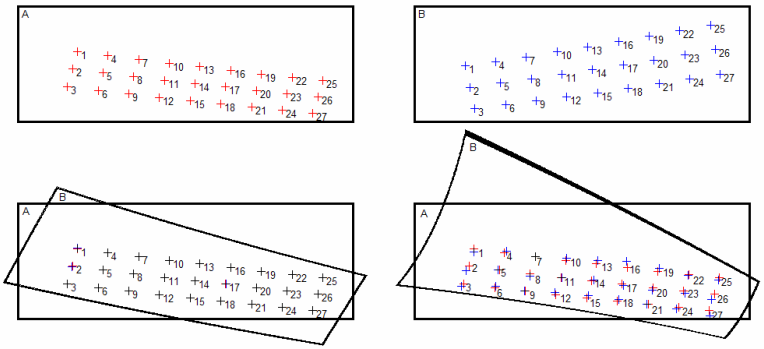



**Fig. 1.** Image A and B represent two linear-pushbroom panoramic images. Only 27 image corresponding points (instead of 250) are shown here for clarity. The bottom figures illustrate the stitching results of two cases: input noise free (left) and noise up to two pixels (right).

**Table 1.** The image correspondence errors (in different noise scale) vs. the image stitching errors (the average square norm).

| **Noise (pixel)** | 0 | 0.25 | 0.5 | 0.75 | 1 | 1.25 | 1.5 | 1.75 | 2 |
|---|---|---|---|---|---|---|---|---|---|
| **Error (pixel)** | 0.05 | 0.69 | 0.82 | 1.37 | 1.77 | 2.34 | 2.49 | 2.66 | 3.48 |

Moreover, a real image example is given in figure 2. The painting is named "Lang Shih-Ling One Hundred Stallions". Sony DCR-VX2000 camera was used and only the central image column of each shot was employed for generating panoramic image. Two linear-pushbroom panoramic images of a small portion of the painting were acquired with certain overlapping, which are shown in figure 2 (A) and (B). The resolutions of these two panoramas are both 450 × 2000 pixels. Figure 2 (C) shows the image stitching result based on 53 identified image corresponding points. The resulting image after Euclidean reconstruction is shown in figure 2 (D), which has resolutions 600 × 1700 pixels. This width/height ratio has been adjusted to meet the true ratio of the selected portion.

## 5 Conclusion

Planar homography, which can help determine complete point-to-point image relationships for a pair of images taken with perspective cameras, serves as a critical property for realizing many image mosaicing applications. Nevertheless, whether similar useful properties exist for other imaging model, such as the linear-pushbroom model, has not been well studied yet. In this paper, we demonstrate that with additional planarity constraint to the scene geometry, complete point-to-point image relationships can also be established between two linear-pushbroom panoramic images by employing at least five pairs of corresponding

**Fig. 2.** Portion of painting "Lang Shih-Ling One Hundred Stallions".

points. By the existence of such property, an image stitching method is developed for integrating two LP-panoramas to enlarge the panorama's field of view. Moreover, a Euclidean reconstruction method is presented to restore the properties of 2D Euclidean geometry for reconstructing a rectangular frame. Both methods required only few pairs of image corresponding points as the input. The image integration algorithm as a whole can be performed in real-time.

## References

1. Chen, S. E.: QuickTime VR - An Image-Based Approach to Virtual Environment Navigation. Computer Graphics (SIGGRAPH'95) (1995) 29-38
2. Chen, S., Williams, L.: View Interpolation for Image Synthesis. Computer Graphics (SIGGRAPH'93) (1993) 279-288
3. Faugeras, O., Luong, Q.-T.: The Geometry of Multiple Images. The MIT Press, London, England (2001)
4. Gortler, S. J., Grzeszczuk, R., Szeliski, R., Cohen, M. F.: The Lumigraph. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH (1996) 43-54
5. Gupta, R., Hartley, R. I.: Linear Pushbroom Cameras. IEEE PAMI, **19**(9), (1997) 963-975
6. Hansen, M., Anadan, P., Dana, K., van de Wal, G., Burt, P.: Realtime Scene Stabilization and Mosaic Construction. In Proc. of IEEE CVPR (1994) 54-62
7. Hartley, R. I.: Multiple View Geometry in Computer Vision, Cambridge University Press (2000)
8. Irani, M., Anandan, P., Hsu, S.: Mosaic Based Representation of Video Sequences and Their Applications. IEEE Proc. Int'l Conf. Computer Vision (1995) 605-611
9. Ishiguro, H., Yamamoto, M., Tsuji, S.: Omni-Directional Stereo. IEEE PAMI, **14**(2) (1992) 257-262
10. Kang, S.: A Survey of Image-Based Rendering Techniques. Videometric VI, **3641** (1999) 2-16
11. Kumar, R., Anandan, P., Hanna, K.: Shape Recovery from Multiple Views: A Parallax Based Approach. In Image Understanding Workshop, Monterey, CA. Morgan Kaufmann Publishers (1994) 947-955

12. Kumar, R., Anandan, P., Irani, M., Bergen, J., Hanna, K.: Representation of Scenes from Collections of Images. In Proc. IEEE Workshop on Representations of Visual Scenes (1995) 10-17
13. Levoy, M., Hanrahan, P.: Light Field Rendering. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH (1996) 31-42
14. Mann, S., Picard, R. W.: Virtual Bellows: Constructing High Quality Stills from Video. In ICIP (1994)
15. McMillan, L., Bishop, G.: Plenoptic Modeling: An Image-Based Rendering System. Computer Graphics (SIGGRAPH'95) (1995) 39-46
16. Moffitt, F. H., Mikhail, E. M.: Photogrammetry. Harper & Row, New York, 3rd Edition (1980)
17. Peleg, S., Herman, J.: Panoramic Mosaics by Manifold Projection. IEEE CVPR Proceedings (1997)
18. Sawhney, H. S.: Simplifying Motion and Structure Analysis Using Planar Parallax and Image Warping. In 12th International Conference on Pattern Recognition (ICPR'94), Vol. A. Jerusalem, Israel. IEEE Computer Society Press (1994) 403-408
19. Shum, H. Y., He, L. W.: Rendering with Concentric Mosaic. SIGGRAPH'99 (1999) 299-306
20. Shum, H. Y., Szeliski, R.: Panoramic Image Mosaics. Tech. Rep. MSR-TR-97-23, Microsoft Research (1997)
21. Szeliski, R.: Image Mosaicing for Tele-Reality Applications. Technical Report 94/2, Digital Equipment Corporation, Cambridge Research Lab (1994)
22. Szeliski, R., Kang, S. B.: Direct Methods for Visual Scene Reconstruction. In IEEE Workshop on Representations of Visual Scenes, Cambridge, Massachusetts (1995) 26-33

## Appendix

Let $E$ and $E'$ be two LP-mosaic images of a common planar scene. We explain and illustrate when the two line sensors used to grab the two LP-mosaic images are parallel to each other, the following statement holds: for any pair of image corresponding points $(u_i, v_i)^T$ and $(u'_i, v'_i)^T$, the values $u_i$ and $u'_i$ are related by the equation: $u_i = Au'_i + B$, where $A$ and $B$ are constants.

First, consider a plane in 3D and a line camera which moves along a straight line (set it to be the x-axis of the camera coordinate system) with constant velocity and taking line images at each position $\mathbf{C}_0$, $\mathbf{C}_1$, $\mathbf{C}_2$, and so on, as shown in the top-left of figure 3. The y-axis is defined to be parallel to the line-sensors and is perpendicular to the x-axis. The z-axis is defined following the right-hand-rule. The geometric relationship between the plane and the camera coordinate system is unknown.

The bottom-left of figure 3 shows the resulting LP-panoramic image $E$. The parallel lines $L_0 \sim L_4$ on the plane are projected to image columns $u = 0 \sim 4$ respectively. Since the distance between any pair of points $\mathbf{C}_i$ and $\mathbf{C}_{i+1}$ is constant (as defined in Section 2), lines $L_i$ and $L_{i+1}$ is a set of parallel lines with equal distance.

Then, consider another LP-panoramic image $E'$, whose associated camera's moving path is rotated with respect to the y-axis, as shown in the right-hand-side of figure 3. The equal-distance parallel lines $L'_0 \sim L'_4$ on the plane are projected to image columns $u' = 0 \sim 4$ respectively.

In fact, lines $L_0 \sim L_4$ are parallel to lines $L'_0 \sim L'_4$ as they both are parallel to the y-axis of the camera coordinate system. So, it is possible that lines $L_0 \sim L_4$ also appear in

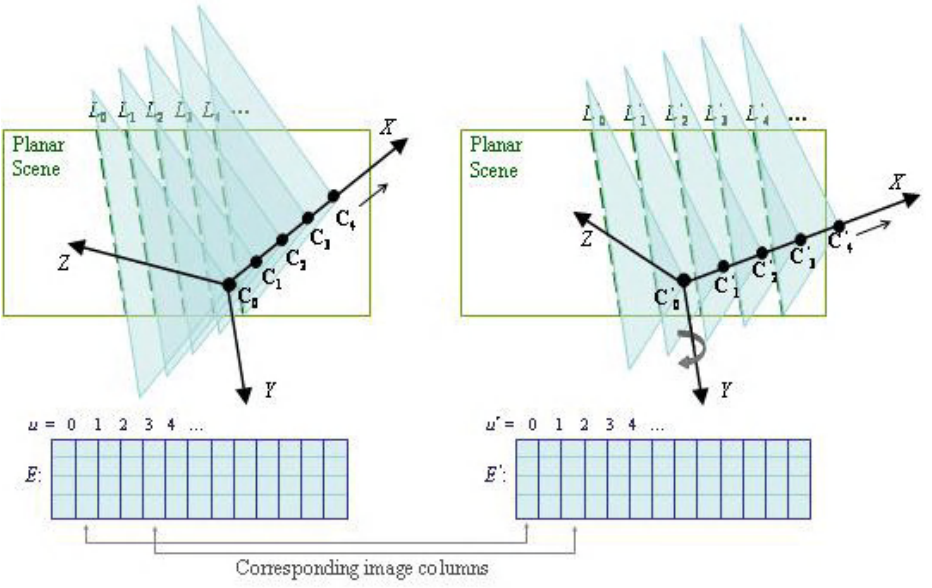



**Fig. 3.** Geometric configuration that illustrates the existence of image column-to-column correspondence.

the image $E'$ and vice versa. Hence, we have column-to-column correspondence between two LP-panoramic images. According to the basic geometrical property, when there is a column-to-column correspondence between two images as described, the relationship between those corresponding columns can be expressed by equation $u_i = Au'_i + B$, where $A$ and $B$ are constants.

Finally, we may conclude that as long as the y-axes of two camera coordinate systems, which are associated to the different LP-panoramic images, are parallel, we have relation $u_i = Au'_i + B$ for all corresponding $u_i$ and $u'_i$.

# Audio-Video Integration for Background Modelling

Marco Cristani, Manuele Bicego, and Vittorio Murino

Dipartimento di Informatica, University of Verona
Ca' Vignal 2, Strada Le Grazie 15, 37134 Verona, Italy
{cristanm,bicego,murino}@sci.univr.it

**Abstract.** This paper introduces a new concept of surveillance, namely, audio-visual data integration for background modelling. Actually, visual data acquired by a fixed camera can be easily supported by audio information allowing a more complete analysis of the monitored scene. The key idea is to build a multimodal model of the scene background, able to promptly detect single auditory or visual events, as well as simultaneous audio and visual foreground situations. In this way, it is also possible to tackle some open problems (e.g., the sleeping foreground problems) of standard visual surveillance systems, if they are also characterized by an audio foreground. The method is based on the probabilistic modelling of the audio and video data streams using separate sets of adaptive Gaussian mixture models, and on their integration using a coupled audio-video adaptive model working on the frame histogram, and the audio frequency spectrum. This framework has shown to be able to evaluate the time causality between visual and audio foreground entities. To the best of our knowledge, this is the first attempt to the multimodal modelling of scenes working on-line and using one static camera and only one microphone. Preliminary results show the effectiveness of the approach at facing problems still unsolved by only visual monitoring approaches.

## 1 Introduction

Automated surveillance systems have acquired an increased importance in the last years, due to their utility in the protection of critical infrastructures and civil areas. This trend has amplified the interest of the scientific community in the field of the video sequence analysis and, more generally, in the pattern recognition area [1]. In this context, the most important low-level analysis is the so called background modelling [2,3], aimed at discriminating the static scene, namely, the background (BG), from the objects that are acting in the scene, i.e., the foreground (FG). Despite the large related literature, there are many problems that are still open [3], like, for instance, the sleeping foreground problem. In general, almost all of the methods work only at the visual level, hence resulting in *video* BG modelling schemes. This could be a severe limitation, since other information modalities are easily available (e.g., audio), which could be effectively used as complementary information to discover "activity patterns" in a scene.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 202–213, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this paper, the concept of *multimodal*, specifically audio-video, BG modelling is introduced, which aims at integrating different kinds of sensorial information in order to realize a more complete BG model. In the literature, the integration of audio and visual cues received a growing attention in the last few years. In general, audio-visual information have been used in the context of speech recognition, and, recently, of scene analysis, especially person tracking. A critical review of the literature devoted to audio-video scene analysis is reported in section 2.

In order to integrate audio and visual information, different adaptive BG mixture models are first designed for monitoring the segregated sensorial data streams. The model for visual data operates at two levels. The first is a typical time-adaptive per-pixel mixture of Gaussians model [2], able to identify the FG present in a scene. The second model works on the FG histogram, and is able to classify different FG events. Concerning the audio processing scheme, the concept of *audio* BG modelling is introduced, proposing a system able to detect unexpected audio events. In short, a multiband frequency analysis was first carried out to characterize the monaural audio signal, by extracting features from a parametric estimation of the power spectral density. The audio BG model is then obtained by modelling these features using a set of adaptive mixtures of Gaussians, one for each frequency subband.

Concerning the on-line fusion of audio information with visual data, the most basic issue to be addressed is the concept of "synchrony", which derives from psycho-physiological research [4,5]. In this work, we consider that visual and audio FG that appear "simultaneously" are synchronous, i.e., likely causally correlated. The correlation augments if both FG events persist along time.

Therefore, a third module based on adaptive mixture models operating on audio-visual data has been devised. This module operates in a hybrid space composed by the audio frequency bands, and the FG histogram bins, allowing the binding of concomitant visual and audio events, which can be labelled as belonging to the same multimodal FG event. In this way, a globally consistent multilevel probabilistic framework is developed, in which the segregated adaptive modules control the different sensorial audio and video streams separately, and the coupled audio-video module monitors the multimodal scenario to detect concurrent events. The three modules are interacting each other to allow a more robust and reliable FG detection.

In practice, our structure of BG modelling is able to face serious issues of standard BG modelling schemes, e.g., the sleeping FG problem [3].

The general idea is that an audio-visual pattern can remain an actual FG even if one of the components (audio or video) is missing. The crucial step is therefore the discovery of the audio-visual pattern in the scene.

In summary, the paper introduces several concepts related to the multimodal scene analysis, discussing the involved problems, showing potentialities and possible future directions of the research. The key contributions of this work are: 1) the definition of the novel concept of multimodal background model, introducing, together with video data, audio information processing performing an

auditory scene analysis using only *one* microphone; 2) a method for integrating audio and video information in order to discover synchronous *audio-visual* patterns *on-line*; 3) the implementation of these audio-visual fusion principles in a probabilistic framework working on-line and able to deal with complex issues in video-surveillance, i.e., the sleeping foreground problem.

The rest of the paper is organized as follows. In Section 2, the state of the art related to the audio-video fusion for scene analysis is presented. The whole strategy is proposed in Section 3, and preliminary experimental results are reported in Section 4. Finally, in Section 5, conclusions are drawn.

## 2 State of the Art of the Audio-Visual Analysis

In the context of audio-visual data fusion it is possible to individuate two principal research fields: the on-line audio-visual association for tracking tasks, and the more generic off-line audio-visual association, in which the concept of audio-visual synchrony is particularly stressed.

In the former, the typical scenario is an indoor known environment with moving or static objects that produce sounds, monitored with fixed cameras and fixed acoustic sensors. If an entity emits sound, the system provides a robust multimodal estimate of the location of the object by utilizing the time delay of the audio signal between the microphones and the spatial trajectory of the visual pattern [6,7]. In [6], the scene is a conference room equipped with 32 omnidirectional microphones and two stereo cameras, in which a multi-object 3D tracking is performed. With the same environmental configuration, in [8] an audio source separation application is proposed: two people speak simultaneously while one of them moves through the room. Here the visual information strongly simplifies the audio source separation.

The latter class of approaches employs only one microphone. In this case the explicit notion of the spatial relationship among sound sources is no more recoverable, so the audio-visual localization process must depend purely on the concept of synchrony, as stated in [9]. Early studies about audio-visual synchrony comes from the cognitive science. Simultaneity is one of the most powerful cues available for determining whether two events define a single or multiple objects; moreover, psychophysical studies have shown that the human attention focuses preferably on sensory information perceived coupled in time, suppressing the others [4]. Particular effort is spent in the study of the situation in which the inputs arrive through two different sensory modalities (such as sight and sound) [5].

Most of the techniques in this context make use of measures based on the mutual information criterion [8,10]. These methods extract the pixels of the video sequences that are most related to the occurring audio data using maximization of the mutual information between the entire audio and visual signals, resulting therefore in an off-line processing. For instance, they are used for video-conference annotation [10]: audio and video features are modelled as Gaussians processes, without a distinction between FG and BG. The association is exploi-

ted by searching for a correlation in time of each pixel with each audio feature. The main problem is that it assumes that the visual pattern remains fixed in space; further, the analysis is carried out completely off-line.

The method proposed in this paper tries to bridge these two research areas. To the best of our knowledge, the proposed system constitutes the first attempt to design an on-line *integrated audio-visual* BG modelling scheme using only one microphone, and working in a loosely constrained environment.

## 3 The Audio-Video Background Modelling

The key methodology is represented by the on-line time-adaptive mixture of Gaussians method. This technique has been used in the past to detect changes in the grey level of the pixels for background modelling purposes [2]. In our case, we would like to exploit this method to detect *audio foreground*, video foreground objects, and *joint audio-video* FG events, by building a robust and reliable *multimodal* background model. The basic concepts of this approach are summarized in Section 3.1. The customization in the case of visual and audio background modelling is presented in Section 3.2, and in Section 3.3, respectively. Finally, the integration between audio and video data is detailed in Section 3.4, and how the complete system is used to solve a typical problem of visual surveillance system is reported in Section 3.5.

### 3.1 The Time-Adaptive Mixture of Gaussians Method

The Time-Adaptive mixture of Gaussians method aims at discovering the deviance of a signal from the expected behavior in an on-line fashion. A typical video application is the well-know BG modelling scheme proposed in [2]

The general method models a temporal signal with a time-adaptive mixture of Gaussians. The probability to observe the value $z^{(t)}$, at time $t$, is given by:

$$P(z^{(t)}) = \sum_{r=1}^{R} w_r^{(t)} \mathcal{N}\left(z^{(t)} | \mu_r^{(t)}, \sigma_r^{(t)}\right) \tag{1}$$

where $w_r^{(t)}$, $\mu_r^{(t)}$ and $\sigma_r^{(t)}$ are the mixing coefficients, the mean, and the standard deviation, respectively, of the $r$-th Gaussian of the mixture associated to the signal at time $t$. At each time instant, the Gaussians in a mixture are ranked in descending order using the $w/\sigma$ value. The $R$ Gaussians are evaluated as possible match against the occurring new signal value, in which a successful match is defined as a pixel value falling within $2.5\sigma$ of one of the component. If no match occurs, a new Gaussian with mean equal to the current value, high variance, and low mixing coefficient replaces the least probable component.

If $r_{hit}$ is the matched Gaussian component, the value $z^{(t)}$ is labelled as unexpected (i.e., foreground) if $\sum_{r=1}^{r_{hit}} w_r^{(t)} > T$, where $T$ is a threshold representing

the minimum portion of the data that supports the "expected behavior". The evolution of the components of the mixtures is driven by the following equations:

$$w_r^{(t)} = (1-\alpha)w_r^{(t-1)} + \alpha M^{(t)}, 1 \leq r \leq R, \tag{2}$$

where $M^{(t)}$ is 1 for the matched Gaussian (indexed by $r_{hit}$), and 0 for the others; $\alpha$ is the adaptive rate that remains fixed along time. It is worth to notice that the higher the adaptive rate, the faster the model is "adapted" to scene changes.

The $\mu$ and $\sigma$ parameters for unmatched Gaussians remain unchanged, but, for the matched Gaussian component $r_{hit}$, we have:

$$\mu_{r_{hit}}^{(t)} = (1-\rho)\mu_{r_{hit}}^{(t-1)} + \rho z^{(t)} \tag{3}$$

$$\sigma_{r_{hit}}^{2\,(t)} = (1-\rho)\sigma_{r_{hit}}^{2\,(t-1)} + \rho \left(z^{(t)} - \mu_{r_{hit}}^{(t)}\right)^T \left(z^{(t)} - \mu_{r_{hit}}^{(t)}\right) \tag{4}$$

where $\rho = \alpha \mathcal{N}\left(z^{(t)} | \mu_{r_{hit}}^{(t)}, \sigma_{r_{hit}}^{(t)}\right)$.

### 3.2 Visual Foreground Detection

One of the goal of this work is to detect untypical video activity patterns starting simultaneously with audio ones. In order to discover these visual patterns, a video processing method has been designed, which is composed by two modules: a standard per-pixel FG detection module, and an histogram-based novelty detection module.

The former is realized using the model introduced in the previous section in a standard way [2], in which the processed signal $z^{(t)}$ is the time evolution of the gray level. We use a set of independent adaptive mixtures of Gaussians, one for each pixel. In this case, an unexpected valued pixel $z_{uv}^{(t)}$ (where $u, v$ are the coordinates of the image pixel) is the visual *foreground*, *i.e.*, $z_{uv}^{(t)} \in FG$. Please, note that all mixtures' parameters are updated with a fixed learning coefficient $\tilde{\alpha}$.

The latter module is also realized using the time-adaptive mixture of Gaussians method, using the same learning rate $\tilde{\alpha}$ of the former module, but in this case we focus on the histogram of those pixels classified as foreground. The idea is to compute at each step the gray level histogram of the FG pixels and associating an adaptive mixture of Gaussian to each bin, looking for variations of the bin's value. This means that we are monitoring the number of pixels of the foreground that have a specific gray value. If the number of pixels associated to the foreground *grows*, i.e., some histogram bins increase their values, then an object is appearing in the scene, otherwise is disappearing. We choose to monitor the histogram instead of the number of FG pixels directly (which can be in principle sufficient to detect new objects), as it allows the discrimination between different FG objects, and in order to detect audio-visual patterns composed by single objects. We are aware that this simple characterization leaves some ambiguities (e.g., two equally colored objects are not distinguishable, even if the impact of this problem may be weakened by increasing the number of bins), but

this representation has the appealing characteristic of being invariant to spatial localization of the foreground, which is not constrained to be statically linked to a spatial location (as in other audio-video analysis approaches)[1].

### 3.3 Audio Background Modelling

The audio BG modelling module aims at extracting information from audio patterns acquired by a *single* microphone. In the literature, several approaches to audio analysis are present, mainly focused on the computational translation of psychoacoustics results. One class of approaches is the so called "computational auditory scene analysis"(CASA) [12], aimed at the separation and classification of sounds present in a specific environment. Closely related to this field, but not so investigated, there is the "computational auditory scene recognition" (CASR) [13,14], aimed at environment interpretation instead of analyzing the different sound sources.

Besides various psycho-acoustically oriented approaches derived from these two classes, a third approach tried to fuse "blind" statistical knowledge with biologically driven representations of the two previous fields, performing audio classification and segmentation tasks [15], and source separation [16,17] (blind source separation). In this last approach, many efforts are devoted in the speech processing area, in which the goal is to separate the different voices composing the audio pattern using several microphones [17] or only one monaural sensor [16].

The approach presented in this paper could be inserted in this last category: roughly speaking, we implement a multiband spectral analysis on the audio signal at video frame rate, extracting energy features from $a_1, a_2, \ldots, a_M$ frequency subbands. More in detail, we subdivide the audio signal in overlapped temporal window of fixed length $W_a$, in which each temporal window ends at the instant corresponding to the $t$-th video frame, as depicted in Fig.1. For each window, a parametric estimation of the power spectral density with the Yule-Walker Auto Regressive method [18] is performed. In this way, an estimation $a_i^{(t)}$ of the spectral energy relative to the interval $[t-W_a, t]$ is obtained for the $i$-th subband, $i = 1, 2, \ldots, M$. These features have been chosen as they are able to discriminate between different sound events [13]; further, they can be easily computed at an elevate temporal rate.

As typically considered [16], the energy during time in different frequency bands can transport independent information. Therefore, we instantiate one time-adaptive mixture of Gaussians for each band of the frequency spectrum. Also in this case, all mixtures' parameters are updated with a fixed learning coefficient $\tilde{\alpha}$, equal to the one used for the video channel. In this way, we are able to discover unexpected audio behaviors for each band, indicating an audio foreground.

[1] Actually, more sophisticated tracking approaches based on histograms have already been proposed in literature [11], and are subjects of future work.

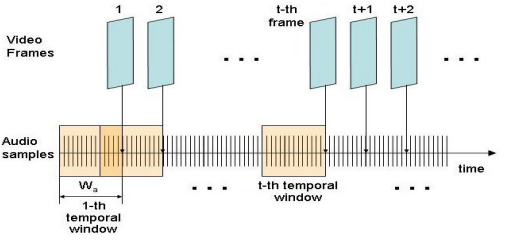



**Fig. 1.** Organization of the multimodal data set: at each video frame, an audio signal analysis is carried out using a temporal window of length $W_a$

### 3.4 The Audio-Visual Fusion

The audio and visual spaces are now partitioned in different independent subspaces, the audio subbands $a_1, a_2, \ldots, a_M$, and the video FG histogram bins $h_1, h_2, \ldots, h_N$, respectively, in which independent FG monomodal patterns may occur. Therefore, given an audio subband $a_i^{(t)}$, and a video histogram bin $h_j^{(t)}$ at time $t$, we can define an history of the mono-modal FG patterns $A_i^{(t)}$, $i = 1, \ldots, M$, and $H_j^{(t)}$, $j = 1, \ldots, N$, as the patterns in which the values of a given component of the $i-th$ mixture for the audio, and the $j-th$ mixture for the video are detected as foreground along time. Formally, let us denote $A_i^{(t)}$ and $H_j^{(t)}$ as:

$$A_i^{(t)} = [a_i^{(t_{q,i})}, a_i^{(t_{q,i}+1)}, \ldots, a_i^{(t)} \in FG] \tag{5}$$

$$H_j^{(t)} = [h_j^{(t_{u,j})}, h_j^{(t_{u,j}+1)}, \ldots, h_j^{(t)} \in FG] \tag{6}$$

where $t_{q,i}$ is the first instant at which the $q-th$ Gaussian component of the audio mixture of the $i-th$ sub-band becomes FG, and the same applies for $t_{u,j}$ related to the video data. Clearly, $A_i^{(t)}$ and $H_j^{(t)}$ are possibly not completely overlapped, so $t_{q,i}$ in general can be different from $t_{u,j}$. Therefore, in order to evaluate the degree of concurrency, we define a *concurrency* value as $\beta_{i,j} = |t_{q,i} - t_{u,j}|$. Obviously, the higher this value, the weaker the synchronization.

As previously stated, the synchronism gives a natural *causal* relationship for processes coming from different modalities [4]. In order to evaluate this causal dependency along time, we state as highly correlated those concurrent audio-video FG patterns explaining, in their jointly evolution, a nearly stable behavior. Consequently, we couple all the audio FG values with all the visual FG values occurring at time step $t$, building an $M \times N$ *audio-visual FG matrix* $AV^{(t)}$, where

$$AV^{(t)}(i,j) = \begin{cases} (a_i^{(t)}, h_j^{(t)}) & \text{if } \; a_i^{(t)} \in FG \bigwedge h_j^{(t)} \in FG \\ empty & \text{otherwise} \end{cases} \tag{7}$$

This matrix gives a snapshot of the degree of synchrony between audio and visual FG values, for all $i, j$. If $AV^{(t)}(i,j)$ is not empty, probably, $A_i^{(t)}$ and $H_i^{(t)}$ are in

some way synchronized. In this last case, we choose to model the evolution of these values using an on-line 2D adaptive Gaussian model. Therefore, at each time step $t$, we can evaluate the probability to observe a pair of audio-visual FG events, $AV^{(t)}(i,j)$, as

$$P(AV^{(t)}(i,j)) = \sum_{r=1}^{R} w_{AV_r}^{(t,i,j)} \mathcal{N}\left(AV^{(t)}(i,j) | \mu_r^{(t,i,j)}, \mathbf{\Sigma}_r^{(t,i,j)}\right) \tag{8}$$

Intuitively, the higher the value of the weight $w_{AV_r}^{(t,i,j)}$ matched by the observation $(a_i^{(t)}, h_j^{(t)})$ at time t, namely $w_{AV_{r_{hit}}}^{(t,i,j)}$, the more stable are the coupled audio-visual FG values along time, and it is more probable that a causal relation is present between audio and visual FG.

All the necessary information to assess the synchrony and the stability of a pair of audio and video FG patterns is now available. Therefore, a modulation of the evolution process of the 2D Gaussian mixture model is introduced in order to give more importance to a match with a couple of FG values belonging to likely synchronized audio and video patterns. We would like to impose that the higher the concurrency, the faster the stability of an AV value must be highlighted. In formulas, omitting the indices $i, j$ for clarity

$$w_{AV_r}^{(t)} = (1 - \alpha_{AV}) w_{AV_r}^{(t-1)} + \alpha_{AV} M_{AV}^{(t)}, \quad 1 \leq r \leq R, \tag{9}$$

where

$$M_{AV}^{(t)} = \begin{cases} \frac{1}{\beta_{i,j}+1} = \frac{1}{|t_q - t_u|+1} & \text{for the matched 2D Gaussian} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

This equation [2] implies that if the synchronization does not occur at the same instant, the weight grows more slowly, and viceversa.

In order to subsume the concurrency and the stability behavior of the multimodal FG patterns, we finally introduce the *causality matrix* $\Gamma^{(t)} = [\gamma_{i,j}^t]$, for all $i = 1, \ldots, M$, and $j = 1, \ldots, N$, where

$$\gamma^{(t)}(i,j) = w_{AV_{r_{hit}}}^{(t,i,j)} \tag{11}$$

where $w_{AV_{r_{hit}}}^{(t,i,j)}$ is the weight of the 2D Gaussian component of the model matched by the pair of FG values $(a_i^{(t)}, h_j^{(t)})$.

As we will see in the experimental session, this model well describe the stability degree of the audio-visual FG, in an on-line unsupervised fashion.

### 3.5 Application to the Sleeping Foreground Problem

The sleeping foreground problem occurs when a moving object, initially detected as foreground, stops, and becomes integrated in the background model after a

[2] Any function inversely proportional to $\beta_{i,j}$ could be used; actually, different function choices do not sensibly affect the method performances.

certain period. We want to face this situation, under the hypothesis that there is a multimodal FG pattern, i.e. detecting the correlation between audio and video FG. In this situation, we maintain as foreground both the visual appearance of the object and the audio pattern detected, as long as they are present and stable in time. Technically speaking, we compute the learning rate of the mixture of Gaussians associated to the video histogram's bin $j$

$$\alpha_j^{(t)} = \min\left(\tilde{\alpha}, 1 - \max_i \gamma^{(t)}(i,j)\right) \tag{12}$$

where $\tilde{\alpha}$ is the learning rate adopted for both the segregated sensorial channels. The learning rates of the adaptive mixtures of all pixels which gray level belongs to the histogram bin $j$ become $\alpha_j^{(t)}$. Moreover, also the learning rate of the mixture associated to the band $\arg\max_i \gamma^{(t)}(i,j)$ becomes $\alpha_j^{(t)}$. This measure implies that the most correlated audio FG pattern with the $j-th$ video FG pattern guides the evolution step, and viceversa. In practice we can distinguish $\min(M,N)$ different audio-video patterns. This may appear a weakness of this method, but this problem may be easily solved by using a finer discretization of the audio spectral, and of the histogram spaces. Moreover, other features could be used for the video data modelling, like, for instance, color characteristics.

## 4 Experimental Results

An indoor audio-visual sequence is considered, in which two sleeping FG situations occur: the former is associated with audio cues, and the latter is not. We will show that our system is able to deal with both situations.

More in detail, the sequence is captured at 30 frames per second, and the audio signal is sampled at 22.050 Hz. The temporal window used for multiband frequency analysis is equal to 1 second, and the order of the autoregressive model is 40. We undersample the $128 \times 120$ video image in a grid of $32 \times 30$ locations. Finally, we use 12 bins for the FG color histogram. Analogously, we perform spectral analysis using $M = 16$ logarithmic spaced frequency subbands, in which the frequency is measured in radians in the range $[0, \pi]$, and the power is measured in Decibel. As a consequence, we have an audio-visual space quantized in $M \times N = 16 \times 12$ elements. All adaptive mixtures are composed by 4 Gaussian components, and the learning parameter for the AV mixtures is fixed to 0.05, and for the separated channels $\tilde{\alpha}$=0.005, initially.

We compare our results with those proposed by an "only video" BG modelling, choosing as reference the standard video BG modelling adopted in [2], showing: 1) the resulting analysis of both BG modelling schemes; 2) the audio BG modelling analysis; 3) the histogram FG modelling analysis, able to individuate the appearance of new visual FG in the scene, and 4) the causality matrix, ordered by audio subbands per video histogram bins, that explains intuitively the intensity causal relationship in the joint audio-visual space.

As one can observe in Fig.2, at frame 50 both per-pixel BG modelling schemes locates a FG entering in the scene. This causes a strong increment in the gray

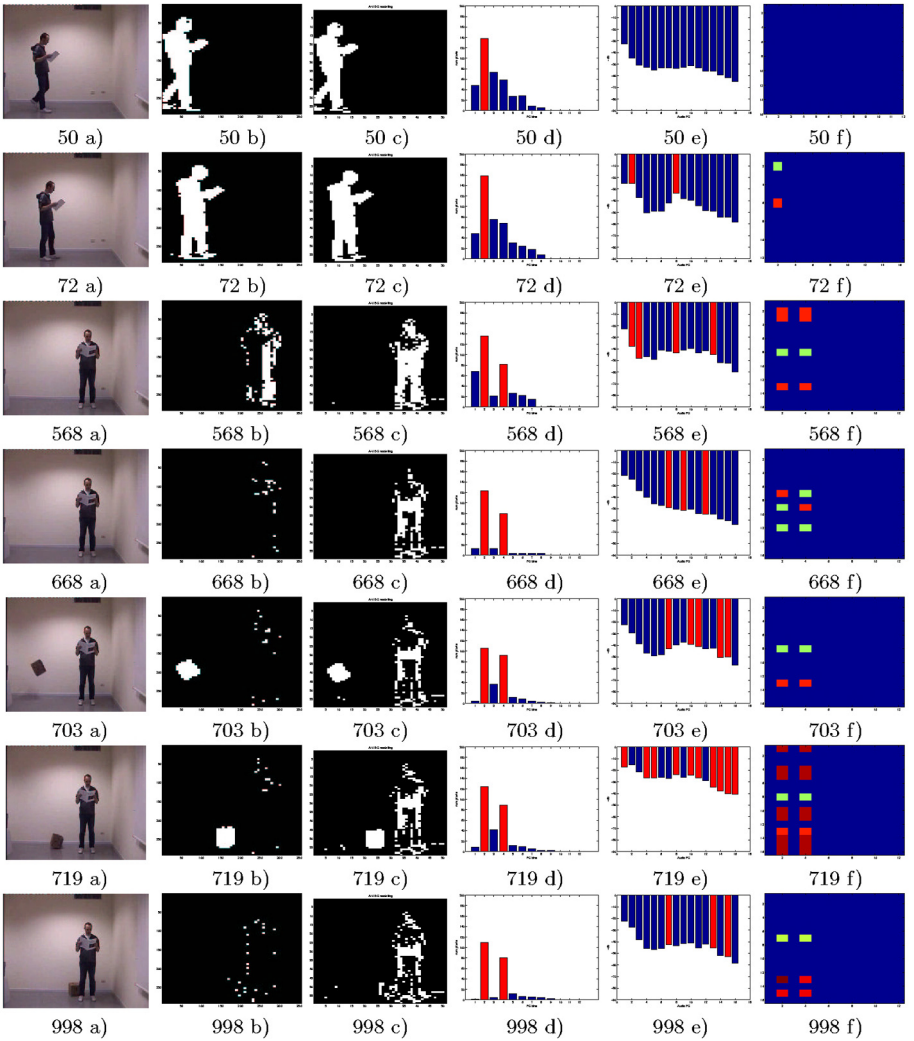



**Fig. 2.** Comparative results: a) Original sequence; b) Ordinary per pixel video BG modelling; c) Our approach; d) Video novelty detection; e) Audio background modelling; f) Causality matrix at time $t$;

level of the FG histogram that correctly detects this object as new (Fig.2-50 d) (the lighter bins indicate FG). At frame 72, the person begins to speak, causing an increment of some subbands of the audio spectrum, which is detected as FG by the audio module (Fig. 2-72 e)). Due to the (loose) synchrony of the audio and visual events, the causality matrix evidences a concurrency, as depicted in Fig. 2-72 e). Here, the lightest colored value indicates $\max_i \gamma^{(t)}(i,j)$, i.e., the

maximum causality relation for all audio subbands $i$, given the video histogram bin $j$. Therefore, proportionally to the temporal stability of the audio-video FG values, the causality matrix increments some of its entries. Consequently, the learning coefficients of the corresponding audio, histogram, and pixels FG models, become close to zero according to eq. 12. In this way, the synchronized audio and visual FG which remain jointly similar along time are considered as multimodal FG. In the typical video BG modelling scheme, if the visual FG remains still in the scene for a lot of iterations (Fig. 2-568 a) and 668 a)), it loses all its meaning of novelty, so becoming assimilated in the background (Fig. 2- 568 b) and 668 b)). More correctly, in the multimodal case, the FG loses its meaning of novelty only if it remains still without producing sound. In Fig. 2-568 c) and 668 c), the visual aspect of the FG is maintained from the audio FG signal, by exploiting the causality matrix.

The audio visual fusion is also able to preserve the adaptiveness of the BG modelling, if the case. In Fig.2-703 a) and 719 a), a box falls near the talking person, providing new audio and video FG, but, after a while, the box becomes still and silent. In this case, it is correct that it becomes BG after some time (see Fig. 2- 998 b). Also in our approach, the box becomes BG, as the audio pattern decreases quickly, so that no audio visual coupling occurs, and after some iterations the box vanishes, whereas the talking person remains detected (Fig.2- 719 c) and 998 c)). A subtle drawback is notable in Fig.2- 998 c): some parts of box do not completely disappears, because their gray level is similar to that of the talking person, modelled as FG. But this problem could be faced by using a different approach to model visual data (instead of the histogram), or, for instance, a finer quantization of the video histogram space.

## 5 Conclusions

In this paper, a new concept of multimodal background modelling has been introduced, aimed at integrating audio and video cues for a more robust and complete scene analysis. The separate audio and video streams are modelled using a set of adaptive Gaussian models, able to discover audio and video foregrounds. The integration of audio and video data is obtained posing particular attention to the concept of synchrony, represented using another set of adaptive Gaussian models. The system is able to discover concurrent audio and video cues, which are bound together to define audio-visual patterns. The integrated probabilistic system is able to work on-line using only one camera and one microphone. Preliminary experimental results have shown that this integration permits to face some problems of still video surveillance systems, like the FG sleeping problem.

**Acknowledgment.** This work was partially supported by the European Commission under the project no. GRD1-2000-25409 named ARROV.

## References


1. PAMI: Special issue on video surveillance. IEEE Trans. on Pattern Analysis and Machine Intelligence **22** (2000)
2. Stauffer, C., Grimson, W.: Adaptive background mixture models for real-time tracking. In: Int. Conf. Computer Vision and Pattern Recognition. Volume 2. (1999)
3. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: Principles and practice of background maintenance. In: Int. Conf. Computer Vision. (1999) 255–261
4. Niebur, E., Hsiao, S., Johnson, K.: Synchrony: a neuronal mechanism for attentional selection? Current Opinion in Neurobiology (2002) 190–194
5. Stein, B., Meredith, M.: The Merging of the Senses. MIT Press, Cambridge (1993)
6. Checka, N., Wilson, K.: Person tracking using audio-video sensor fusion. Technical report, MIT Artificial Intelligence Laboratory (2002)
7. Zotkin, D., Duraiswami, R., Davis, L.: Joint audio-visual tracking using particle filters. EURASIP Journal of Applied Signal Processing **2002** (2002) 1154–1164
8. Wilson, K., Checka, N., Demirdjian, D., Darrell, T.: Audio-video array source separation for perceptual user interfaces. In: Proceedings of Workshop on Perceptive User Interfaces. (2001)
9. Darrell, T., Fisher, J., Wilson, K.: Geometric and statistical approaches to audio-visual segmentation for unthetered interaction. Technical report, CLASS Project (2002)
10. Hershey, J., Movellan, J.R.: Audio-vision: Using audio-visual synchrony to locate sounds. In: Advances in Neural Information Processing Systems 12, MIT Press (2000) 813–819
11. Mason, M., Duric, Z.: Using histograms to detect and track objects in color video. In: The 30th IEEE Applied Imagery Pattern Recognition Workshop (AIPR'01), Washington, D.C., USA (2001) 154–159
12. Bregman, A.: Auditory Scene Analysis: The Perceptual Organization of Sound. MIT Press, London (1990)
13. Peltonen, V.: Computational auditory scene recognition. Master's thesis, Tampere University of Tech., Finland (2001)
14. Cowling, M., R.Sitte: Comparison of techniques for environmental sound recognition. Pattern Recognition Letters (2003) 2895–2907
15. Zhang, T., Kuo, C.: Audio content analysis for online audiovisual data segmentation and classification. IEEE Transactions on Speech and Audio Processing **9** (2001) 441–457
16. Roweis, S.: One microphone source separation. In: Advances in Neural Information Processing Systems. (2000) 793–799
17. Hild II, K., Erdogmus, D., Principe, J.: On-line minimum mutual information method for time-varying blind source separation. In: Intl. Workshop on Independent Component Analysis and Signal Separation (ICA '01). (2001) 126–131
18. Marple, S.: Digital Spectral Analysis. second edn. Prentice-Hall (1987)

# A Combined PDE and Texture Synthesis Approach to Inpainting

Harald Grossauer

Department of Computer Science
University of Innsbruck
Technikerstr. 25
A–6020 Innsbruck
AUSTRIA
harald.grossauer@uibk.ac.at
http://informatik.uibk.ac.at/infmath/

**Abstract.** While there is a vast amount of literature considering PDE based inpainting and inpainting by texture synthesis, only a few publications are concerned with combination of both approaches. We present a novel algorithm which combines both approaches and treats each distinct region of the image separately. Thus we are naturally lead to include a segmentation pass as a new feature. This way the correct choice of texture samples for the texture synthesis is ensured. We propose a novel concept of "local texture synthesis" which gives satisfactory results even for large domains in a complex environment.

## 1 Introduction

The increase in computing power and disk space over the last few decades has created new possibilities for image and movie postprocessing. Today, old photographs which are threatened by bleaching can be preserved digitally. Old celluloid movies, taking more and more damage every time they are exhibited, can be digitized and preserved. Unfortunately much material has already suffered. Typical damages are scratches or stains in photographs, peeled of coatings, or dust particles burned into celluloid. All these flaws create regions where the original image information is lost. Manual restoration of images or single movie frames is possible, but it is desirable to automate this process. Several *inpainting algorithms* have been developed to achieve this goal. In this paper we focus on single image inpainting algorithms (there exist more specialized algorithms for movie inpainting). They may roughly be divided into two categories:

1. Usually PDE based algorithms are designed to connect edges (discontinuities in boundary data) or to extend level lines in some adequate manner into the inpainting domain, see [1,2,3,4,5,6,7,8,9,10,11]. They are targeted on extrapolating geometric image features, especially edges. I.e. they create regions inside the inpainting domain. Most of them produce disturbing artifacts if the inpainting domain is surrounded by textured regions, see figure 1.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 214–224, 2004.
© Springer-Verlag Berlin Heidelberg 2004

2. Texture synthesis algorithms use a sample of the available image data and aim to fill the inpainting domain such that the color relationship statistic between neighbored pixels matches those of the sample, see [12,13,14,15,16, 17,18]. They aim for creating intra–region details. If the inpainting domain is surrounded by differently textured regions, these algorithms can produce disturbing artifacts, see figure 2.

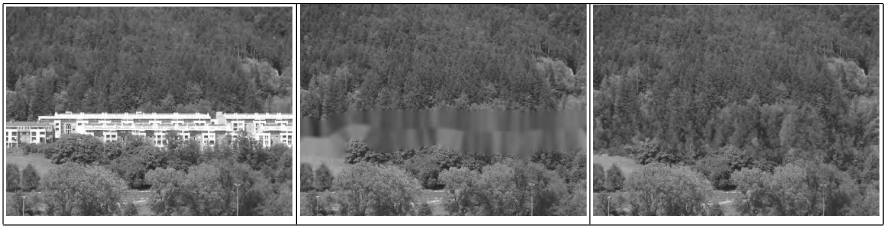


**Fig. 1.** An example which is not well suited for PDE inpainting. The texture synthesis algorithm achieves visually attractive results *(right picture)*, PDE based inpainting algorithms fail for large sized domains surrounded by strongly textured areas *(middle picture)*

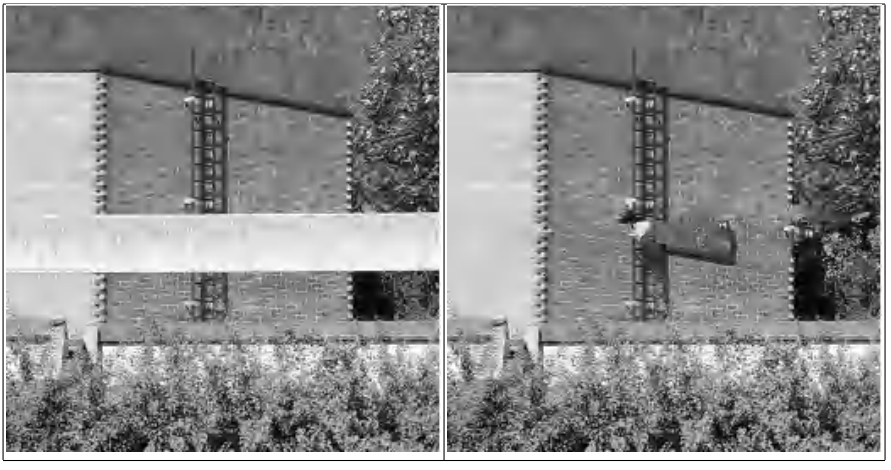


**Fig. 2.** Texture synthesis may run into problems, if the sampling domain is chosen inappropriately. The balustrade in the *left* picture should be removed by resynthesizing image contents, taking the rest of the picture as sample texture. The result can be seen on the *right*: the ladder initiated spurious sampling of trees and leaves into the brick wall

Until now there are only a few algorithms trying to treat geometric image features and texture simultaneously:

- An algorithm based on texture template spectrum matching has been proposed in [19]. This algorithm does not fit into either one of the two categories mentioned above.
- A special purpose algorithm was used in [20] for restoring missing blocks in wireless transmitted compressed images. Common lossy compression algorithms (like e.g. JPEG) divide an image into 8x8 pixel blocks that are independently compressed. If a corrupted block is detected it is reconstructed rather than retransmitted to decrease latency and bandwidth usage. Reconstruction occurs by classifying the contents of adjacent blocks into either structure or texture. Depending on this classification the missing block is restored by invoking either a PDE inpainting or a texture synthesis algorithm.
- Closely related to our inpainting technique are the algorithms proposed in [21,22], which are most natural to compare with in the course of this paper. They differ from our algorithm by the choice of the subalgorithms in each step. Moreover, we propose to perform a segmentation step to determine appropriate texture sample regions. This prevents artifacts arising in the texture synthesis step, as exemplified in figure 2. This problem is not adressed in [21,22].

## 2 The Algorithm

The proposed inpainting algorithm consists of five steps:

1. Filtering the image data and decomposing it into geometry and texture
2. PDE inpainting of the geometry part
3. Postprocessing of the geometry inpainting
4. Segmentation of the inpainted geometry image
5. Synthesizing texture for each segment

We will describe each step in detail in the following subsections. The image, denoted by a function $u : D \to \mathbb{R}$ (or $\mathbb{R}^3$ for color images), is defined on the image domain $D \subset \mathbb{R}^2$. A user specified *mask function* $m : D \to [0, 1]$ marks the inpainting domain $\Omega = \text{supp}(m)$. A value of 1 in the mask function highlights the flawed region. The mask is designated to continuously drop to zero in a small neighborhood (i.e., a few pixels) outside of the flawed region. This "drop down" zone will later be used to smoothly blend the inpainting into the original image.

### 2.1 Filtering and Decomposition

A nonlinear diffusion filter of Perona–Malik type [23] is applied to the image $u$, i.e. it is evolved according the partial differential equation

$$\frac{\partial u}{\partial \tau} = \nabla \cdot (d(\|\nabla u(x, y)\|) \cdot \nabla u) \tag{1}$$

where the diffusivity $d(s)$ is chosen to be

$$d(s) = \frac{1}{1 + \frac{s^2}{\lambda^2}} \tag{2}$$

with a suitably chosen parameter $\lambda$. The filtered image $g$ is the solution of (1) at a specified time $\tau_0$, characterizing the strength of filtering. The effect of the filter is that $g$ reveals piecewise constant intensities and contains little texture and noise. Thus we call $g$ the *geometry part*. We set $u = g + t$ and refer to $t$ as the texture part. All along this paper we consider noise as a texture pattern.

Recently several advanced techniques for image decomposition have been proposed, see [24,25,26]. In [21] decomposition is done by using the model from [24], i.e. by jointly minimizing the BV seminorm of the function $g$ and Meyers G-norm (see [27]) of the function $t$. Using this model allows one to extract texture *without* noise, which leads to an approximative decomposition $u \approx g + t$, with geometry part $g$ and texture $t$. For inpainting this seems not to be optimal, since a noisefree inpainting in a noisy image may look inadequate.

In [22] a lowpass/highpass filter is applied to the image to attain $g$ resp. $t$. The decomposition is thus not into geometry and texture but rather into high and low frequencies.

### 2.2 PDE Inpainting

For the inpainting of the geometry part we use the Ginzburg–Landau algorithm proposed in [11]. Here we give a short overview:

We calculate a complex valued function $\tilde{g}$, whose real part is the geometry image $g$ scaled to a range of values between -1 and 1. Further we demand that $\|\tilde{g}(x,y)\|_{\max} = 1$ for all $(x,y)$, and the imaginary part be nonnegative, $\Im\tilde{g} \geq 0$. The function $\tilde{g}$ is evolved using the *complex* Ginzburg–Landau equation

$$\frac{\partial \tilde{g}}{\partial \tau} = \Delta\tilde{g} + \frac{1}{\varepsilon^2}\left(1 - \|\tilde{g}(x,y)\|^2_{\max}\right)\tilde{g} \tag{3}$$

inside $\Omega$, where the available data $\tilde{g}|_{\partial\Omega}$ is specified as Dirichlet boundary condition. Here $\varepsilon \in \mathbb{R}$ is a length parameter specifying the width of edges in the inpainting. $\|\cdot\|_{\max}$ denotes the maximum norm of the components of $\tilde{g}(x,y)$, which is just the absolute value $|\tilde{g}(x,y)|$ if $g$ is a grayscale image, and $\max\{|\tilde{g}^{red}|, |\tilde{g}^{green}|, |\tilde{g}^{blue}|\}$ for RGB images. The real part $\Re\tilde{g}$ of the evolved image at some time $\tau_0$ (i.e., if $\tilde{g}$ is "close enough" to steady state) is rescaled to the intensity range of $g$ and constitutes the inpainting.

For a more detailed description we refer to our presentation in [11]. Theoretical results about the Ginzburg–Landau equation and similar reaction–diffusion equations can be found in [28,29].

In [21] the PDE inpainting method from [4] is used. In comparison with (3) this algorithm (judging from the examples given in [4]) creates smoother and better aligned edges, but the Ginzburg–Landau algorithm reveals higher contrast and less color smearing.

In [22] the inpainting technique from [10] is utilized. This algorithm is not designed to create edges in the inpainting domain. For the particular application to inpaint a low pass filtered image this is no problem.

### 2.3 Postprocessing

The Ginzburg–Landau algorithm sometimes produces kinks and corners in edges. A detailed discussion of this phenomenon has been given in [11]. This is the price for high contrast edges in the inpainting domain. To straighten the kinks we apply a coherence enhancing anisotropic diffusion filter to the image $g$ on the inpainting domain $\Omega$. This filter is described in [30]. We implemented this diffusion filter using the multi–grid algorithm outlined in [31]. Since diffusion happens mostly along edges and not across, the contrast does not suffer significantly.

### 2.4 Segmentation

As a preparation for the texture synthesizing step the inpainted and postprocessed geometry image (which for the sake of simplicity is again denoted by $g$) is segmented. We employ a gradient controlled region growing algorithm, inspired by the *scalar* Ginzburg–Landau equation:

Let $(\mathcal{S}_i \subseteq D)_{i=0..N}$ be the (a–priori unknown) segmentation of $D$, i.e. $\bigcap_i \mathcal{S}_i = \emptyset$ and $\bigcup_i \mathcal{S}_i \supseteq \Omega$. We assume that every pixel in $\Omega$ belongs to exactly one segment $\mathcal{S}_i$. We do not need to segment the whole image domain $D$ since segments which have no intersection with $\Omega$ do not affect the final result. We derive the sets $\mathcal{S}_i$ from a set of auxiliary functions $(S_i : D \to \mathbb{R})_{i=0..N}$:

1. Set $i = 0$.
2. Choose an arbitrary pixel $(j,k) \in \Omega \setminus \bigcup_{0 \le n < i} \mathcal{S}_n$. Set $S_i = -1$, except for $S_i(j,k) = +1$.
3. Evolve $S_i$ according to the equation

$$\frac{\partial S_i}{\partial t} = \Delta S_i - P'(S_i) - \alpha \|\nabla g(x,y)\| \tag{4}$$

   where $P'(x)$ is the derivative of the polynomial potential

$$P(x) = \frac{9}{4}x^4 + \frac{19}{8}x^3 - \frac{9}{2}x^2 - \frac{57}{8}x \tag{5}$$

   until a steady state is reached.
4. Set $\mathcal{S}_i = \operatorname{supp}\left(\max\left(0, S_i^f\right)\right)$, where $S_i^f$ is the steady state solution from the previous step.
5. If $\bigcup_{0 \le n \le i} \mathcal{S}_n \supseteq \Omega$ terminate the algorithm, else set $i \leftarrow i+1$ and continue with step 2.

Explanation of equation (4): like in the scalar Ginzburg–Landau equation $P'(x)$ is the derivative of a bistable polynomial potential $P(x)$, forcing $S_i(x)$ to take on values close to $+1$ or $-1$. Here $P(x)$ is chosen to be nonsymmetric, having a shallow minimum at $x = -1$ and a deep minimum at $x = +1$. Assume $\alpha = 0$.

$P(x)$ is constructed such that the diffusion caused by the Laplacian is strong enough to move $S_i$ in the surrounding of the seeding pixel from the negative to the positive minimum. Thus under continued evolution the $+1$ domain will spread out all over $D$. If $\alpha > 0$ the term depending on $\nabla g$ is a forcing term acting against diffusion and thus eventually stops propagation at edges of $g$. Eventually further terms could be added, e.g. penalizing curvature of $S_i$ to prevent the segments from crossing small narrow gaps. This turned out not to be required by our application.

Large changes in $S_i$ occur only in a small region between the $+1$ and the $-1$ domain, so this algorithm can be efficiently implemented using a front tracking method. Therefore only a small fraction of all pixels has to be processed at each iteration.

### 2.5 Texture Synthesis

For the texture synthesis step we employed the algorithm from [13]. In [22] the same algorithm with a different implementation has been used. In [21] the texture synthesis algorithm from [18] was used, which should give similar synthesis results as [13]. In both [21,22] all of the available image data $t|_{D\setminus\Omega}$ is taken as sample to synthesize texture in all of $t|_\Omega$. Since the texture synthesis proceeds from the border on inwards into the inpainting domain it is evident that texture can be continued without artifacts. A few "perturbing pixels" might suffice though to make the algorithm use texture sample data from unsuitable image locations, see figure 2.

To circumvent the shortcomings of this "global sampling texture synthesis" we introduce "texture synthesis by local sampling": for every segment $\mathcal{S}_i$ we take $\Omega_i^{sample} = \mathcal{S}_i \setminus \Omega$ and $\Omega_i^{synth} = \mathcal{S}_i \cap \Omega$ to be the texture sample region, resp. the texture synthesizing region. Then the texture synthesis algorithm from [13] is applied to each pair $(\Omega_i^{sample}, \Omega_i^{synth})$ individually. Here we tacitly assume that differently textured regions belong to different segments. Two neighboring regions with different textures belong to the same segment if they have similar intensities, due to the initial diffusion filtering. However, our experiments have shown that the impact of a few wrong texture samples is not significant.

The texture synthesis is the most time consuming part of our inpainting algorithm: for every $(j,k) \in \Omega^{synth}$ set $t_{j,k} \leftarrow t_{m,n}$, where $(m,n) \in \Omega^{sample}$ is chosen such that $t$ in a neighborhood of $(m,n)$ most closely resembles $t$ in a neighborhood of $(j,k)$. Finding the most similar neighborhood for every pixel in $\Omega_{synth}$ leads to a considerable amount of nearest neighbor searches in a high dimensional vector space. In most of our examples the number of *test vectors* (i.e. the number of pixels in $\Omega_{sample}$) was too small – resp. the dimension of the vector space was too high – for a binary search tree to be effective. The runtime of the texture synthesis using a search tree could not be improved compared to an exhaustive search. See [32] for a efficiency discussion of nearest neighbor search algorithms and the presentation of the algorithm that we used.

### 2.6 Assembling

In a last step the synthesized textures are added to the geometry inpainting. The final inpainting is blended onto the initial (flawed) image, i.e. $u_{final} = m \cdot (g + t) + (1 - m) \cdot u_{initial}$ where $m$ is the mask function. This is to soften the impact of discontinuities which could arise from texture synthesis.

## 3 Results and Discussion

The results presented in this section are chosen to highlight various situations an inpainting algorithm has to tackle:

1. In figure 3 the inpainting domain consists of thin and long structures, as occurring in scratch and text removal. This is easier to inpaint than equally sized compact shaped areas, since edges have to be established over short distances only. Additionally, if the size of typical texture features is comparable to the "width" of the inpainting domain, then even inappropriately synthesized texture does not necessarily produce an artifact.
2. In figure 4 inpainting is easy because $\Omega$ is surrounded by a single homogeneous region. The PDE inpainting only needs to adapt the appropriate color. An appropriate texture sampling region is easily found.
3. Inpainting is difficult if the object to be removed covers a variety of different regions containing complex textures, which happens mainly in airbrushing applications, see figure 5.

One difficult example is shown in figure 5: the balustrade to be removed covers three adjacent regions with two different textures. The brick wall is considered as two distinct regions, because of the noticeable difference in brightness. Compared to the plain texture synthesis result from figure 2 no improper texture is synthesized into the wall, due to the local sampling. Unfortunately the corners of the building are found to be another segment and the brick pattern on the corners is not synthesized satisfactorily (neither is it in figure 2). More examples can be found in [33].

### 3.1 Choice of Parameters

Our proposed algorithm contains several numeric parameters that may be tuned: the edge sensitivity $\lambda$ in the pre–filtering, the edge width $\varepsilon$ in the PDE inpainting, an edge sensitivity and a regularization parameter for the post–processing (which have not been explicitly mentioned) and the strength $\alpha$ of the forcing term in the segmentation. Further, for the diffusion equations in the pre– and post–filtering stopping times (resp. stopping cireteria) have to be specified (not for the inpainting and the segmentation, which are evolved to steady state). Moreover, the size and the shapes of the neighbourhood regions in the texture synthesis phase could also be adjusted.

The examples in this paper have been created with a fixed parameter setting that has been tuned on an appropriate training set. We found that the quality of

**Fig. 3.** This example is easy since the inpainting domain consists of long thin structures only

**Fig. 4.** This example is easy because the object to be removed is surrounded by a single weakly textured region

the inpainting was increased only marginally if the parameters were fine–tuned for each image separately. Automatic content based parameter determination resulted in better quality only in a few cases but produced serious artifacts more often. Besides, not all parameters may be chosen independently, i.e., if $\varepsilon$ is increased then so should be $\alpha$.

### 3.2 Future Work

As already pointed out in [21] each separate step of the inpainting algorithm could be performed with several different subalgorithms. Since it is unlikely that one combination performs optimally, it would be desirable to have a criterion for automatically choosing the appropriate algorithms. This criterion would have to include the form of the inpainting domain, image contents, amount of texture and probably more.

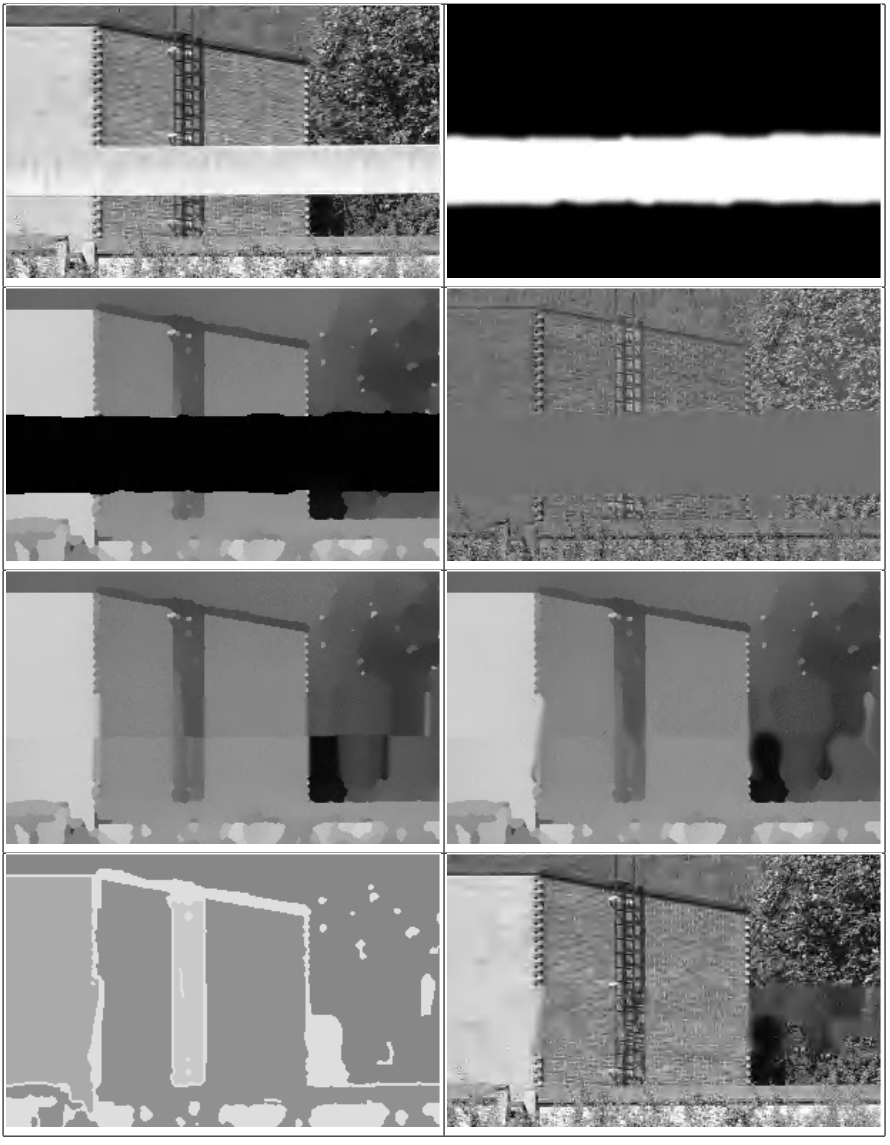


**Fig. 5.** The airbrushed image during each step of the algorithm. **First row**: the original image and the mask, with the inpainting domain being white. **Second row**: the filtered image and the difference image (i.e. the texture part). **Third row**: the inpainted geometry part before (*left*) and after (*right*) postprocessing. **Fourth row**: result of the segmentation and final result of the complete inpainting algorithm. Note that compared to figure 2 textures are synthesized using only appropriate information from the surrounding region. Unfortunately the salient brick pattern on the edge of the building was not correctly recognized by the segmentation

**Acknowledgment.** This work is supported by the Austrian Science Fund (FWF), grants Y–123 INF and P15617-N04.

## References

1. S. Masnou, *Disocclusion: A Variational Approach Using Level Lines*, IEEE Transactions on Signal Processing, 11(2), February 2002, p.68–76
2. S. Masnou, J.-M. Morel, *Level Lines based Disocclusion*, Proceedings of the 1998 IEEE International Conference on Image Processing, p.259–263
3. C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, *Filling–In by Joint Interpolation of Vector Fields and Gray Levels*, IEEE Transactions on Signal Processing, 10(8), August 2001, p.1200–1211
4. M. Bertalmio, G. Sapiro, C. Ballester, V. Caselles, *Image inpainting*, Computer Graphics, SIGGRAPH 2000, July 2000
5. M. Bertalmio, A. Bertozzi, G. Sapiro, *Navier–Stokes, Fluid Dynamics, and Image and Video Inpainting*, IEEE CVPR 2001, Hawaii, USA, December 2001
6. T. Chan, J. Shen, *Mathematical Models for Local Nontexture Inpaintings*, SIAM Journal of Applied Mathematics, 62(3), 2002, p.1019–1043
7. T. Chan, J. Shen, *Non–Texture Inpainting by Curvature–Driven Diffusions (CDD)*, Journal of Visual Communication and Image Representation , 12(4), 2001, p.436–449
8. T. Chan, S. Kang, J. Shen, *Euler's Elastica and Curvature Based Inpainting*, SIAM Journal of Applied Mathematics, 63(2), pp. 564–592, 2002
9. S. Esedoglu, J. Shen, *Digital Inpainting Based on the Mumford–Shah–Euler Image Model*, European Journal of Applied Mathematics, 13, pp. 353–370, 2002
10. M. Oliveira, B. Bowen, R. McKenna, Y. Chang, *Fast Digital Inpainting*, Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain, pp. 261–266
11. H. Grossauer, O. Scherzer, *Using the Complex Ginzburg–Landau Equation for Digital Inpainting in 2D and 3D*, Scale Space Methods in Computer Vision, Lecture Notes in Computer Science 2695, Springer
12. H. Igehy and L. Pereira, *Image replacement through texture synthesis*, Proceedings of the 1997 IEEE International Conf. on Image Processing, 1997
13. Li-Yi Wei, M. Levoy, *Fast Texture Synthesis using Tree-structured Vector Quantization*, Proceedings of SIGGRAPH 2000
14. Li-Yi Wei, M. Levoy, *Order-Independent Texture Synthesis*, Technical Report TR-2002-01, Computer Science Department, Stanford University, April, 2002
15. P. Harrison, *A non–hiearachical procedure for re–synthesis of complex textures*, WSCG'2001, pages 190–197, Plzen, Czech Republic, 2001, University of West Bohemia
16. M. Ashikhmin, *Synthesizing Natural Textures*, Proceedings of 2001 ACM Symposium on Interactive 3D Graphics, Research Triangle Park, NorthCarolina March 19-21, pp. 217-226
17. R. Paget, D. Longstaff, *Texture Synthesis via a Nonparametric Markov Random Field*, Proceedings of DICTA-95, Digital Image Computing: Techniques and Applications, 1, pp. 547–552, 6-8th December 1995
18. A. Efros, T. Leung, *Texture Synthesis by Non-parametric Sampling*, IEEE International Conference on Computer Vision (ICCV'99), Corfu, Greece, September 1999

19. A. Hirani, T. Totsuka, *Combining frequency and spatial domain information for fast interactive image noise removal*, ACM SIGGRAPH, pp 269–276, 1996
20. S. Rane, M. Bertalmio, G. Sapiro, *Structure and Texture Filling-in of missing image blocks for Wireless Transmission and Compression*, IEEE Transactions on Image Processing, 12(3), March 2003
21. M. Bertalmio, L. Vese, G. Sapiro, S. Osher, *Simultaneous texture and structure image inpainting*, IEEE Transactions on Image Processing, 12(8), August 2003
22. H. Yamauchi, J. Haber, H.-P. Seidel, *Image Restoration using Multiresolution Texture Synthesis and Image Inpainting*, Proc. Computer Graphics International (CGI) 2003, pp.120-125, 9-11 July, Tokyo, Japan
23. P. Perona, J. Malik, *Scale–Space and Edge Detection Using Anisotropic Diffusion*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.12, No. 7, July 1990
24. L. Vese, S. Osher, *Modeling Textures with Total Variation Minimization and Oscillating Patterns in Image Processing*, UCLA CAM Report 02-19, May 2002, to appear in Journal of Scientific Computing
25. S. Osher, A. Sole, L. Vese, *Image decomposition and restoration using total variation minimization and the* $H^{-1}$ *norm*, UCLA Computational and Applied Mathematics Reports, October 2002
26. J.-F. Aujol et.al., *Image Decomposition Application to SAR Images*, Scale Space Methods in Computer Vision, Lecture Notes in Computer Science 2695, Springer
27. Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, AMS University Lecture Series 22, 2002
28. F. Bethuel, H. Brezis, F. Helein, *Ginzburg–Landau Vortices*, in Progress in Nonlinear Partial Differential Equations, Vol. 13, Birkhäuser, 1994
29. L. Ambrosio, N. Dancer, *Calculus of Variations and Partial Differential Equations*, Springer, 2000
30. J. Weickert, *Anisotropic Diffusion in Image Processing*, B.G.Teubner, Stuttgart, 1998
31. S. Acton, *Multigrid Anisotropic Diffusion*, IEEE Transactions on Image Processing, 7(3), March 1998
32. S.A. Nene, S.K. Nayar, *A Simple Algorithm for Nearest Neighbor Search in High Dimensions*, IEEE Trans. Pattern Anal. Machine Intell., vol. 19, 989-1003, 1997
33. http://informatik.uibk.ac.at/infmath/

# Face Recognition from Facial Surface Metric

Alexander M. Bronstein[1], Michael M. Bronstein[1],
Alon Spira[2], and Ron Kimmel[2]

[1] Technion - Israel Institute of Technology, Department of Electrical Engineering, 32000 Haifa, Israel
{alexbron,bronstein}@ieee.org

[2] Technion - Israel Institute of Technology, Department of Computer Science, 32000 Haifa, Israel
{ron,salon}@cs.technion.ac.il

**Abstract.** Recently, a 3D face recognition approach based on geometric invariant signatures, has been proposed. The key idea is a representation of the facial surface, invariant to isometric deformations, such as those resulting from facial expressions. One important stage in the construction of the geometric invariants involves in measuring geodesic distances on triangulated surfaces, which is carried out by the *fast marching on triangulated domains* algorithm.
Proposed here is a method that uses only the metric tensor of the surface for geodesic distance computation. That is, the explicit integration of the surface in 3D from its gradients is not needed for the recognition task. It enables the use of simple and cost-efficient 3D acquisition techniques such as photometric stereo. Avoiding the explicit surface reconstruction stage saves computational time and reduces numerical errors.

## 1 Introduction

One of the challenges in face recognition is finding an invariant representation for a face. That is, we would like to identify different instances of the same face as belonging to a single subject. Particularly important is the invariance to illumination conditions, makeup, head pose, and facial expressions – which are the major obstacles in modern face recognition systems.

A relatively new trend in face recognition is an attempt to use 3D imaging. Besides a conventional face picture, three dimensional images carry all the information about the geometry of the face. The usage of this information, or part of it, can potentially make face recognition systems less sensitive to illumination, head orientation and facial expressions.

In 1996, Gordon showed that combining frontal and profile views can improve recognition accuracy [1]. This idea was extended by Beumier and Acheroy, who compared central and lateral profiles from the 3D facial surface, acquired by a structured light range camera [2]. This approach demonstrated some robustness to head orientations. Another attempt to cope with the problem of head pose was presented by Huang et al. using 3D morphable head models [3]. Mavridis

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 225–237, 2004.
© Springer-Verlag Berlin Heidelberg 2004

et al. incorporated a range map of the face into the classical face recognition algorithms based on PCA and hidden Markov models [4]. Their approach showed robustness to large variations in color, illumination and use of cosmetics, and it also allowed separating the face from a cluttered background.

Recently, Bronstein, Bronstein, and Kimmel [5] introduced a new approach which is also able to cope with problems resulting from the non-rigid nature of the human face. They applied the bending invariant canonical forms proposed in [6] to the 3D face recognition problem. Their approach is based on the assumption that most of human facial expressions are near-isometric transformations of the facial surface. The facial surface is converted into a representation, which is invariant under such transformations, and thus yields practically identical signatures for different postures of the same face.

One of the key stages in the construction of the bending invariant representation is the computation of the geodesic distances between points on the facial surface. In [5], geodesic distances were computed using the Fast Marching on Triangulated Domains (FMTD) algorithm [7]. A drawback of this method is that it requires a polyhedral representation of the facial surface. Particularly, in [5] a coded-light range camera producing a dense range image was used [9]. Commercial versions of such 3D scanner are still expensive.

In this paper, we propose 3D face recognition based on simple and cheap 3D imaging methods, that recover the local properties of the surface without explicitly reconstructing its shape in 3D. One example is the photometric stereo method, that first recovers the surface gradients. The main novelty of this paper is a variation of the FMTD algorithm, capable of computing geodesic distances given only the metric tensor of the surface. This enables us to avoid the classical step in shape from photometric stereo of integrating the surface gradients into a surface.

In Section 2 we briefly review 3D imaging methods that recover the metric tensor of the surface before reconstructing the surface itself; Section 3 is dedicated to the construction of bending-invariant canonical forms [6], and in Section 4 we present our modified FMTD algorithm. Section 5 shows how 3D face recognition works on photometric stereo data. Section 6 concludes the paper.

## 2 Surface Acquisition

The face recognition algorithm discussed in this paper treats faces as three-dimensional surfaces. It is therefore necessary to obtain first the facial surface of the subject that we are trying to recognize.

Here, our main focus is on 3D surface reconstruction methods that recover *local* properties of the facial surface, particularly the surface gradient. [1] As we will show in the following sections, the actual surface reconstruction is not really needed for the recognition.

[1] The relationship between the surface gradient and the metric tensor of the surface is established in Section 4 in equations (16) and (18).

### 2.1 Photometric Stereo

The photometric stereo technique consists of obtaining several pictures of the same subject in different illumination conditions and extracting the 3D geometry by assuming the Lambertian reflection model. We assume that the facial surface, represented as a function, is viewed from a given position along the z-axis. The object is illuminated by a source of parallel rays directed along $l^i$ (Figure 1).

$$I^i(x,y) = \max(\rho(x,y)n(x,y)\cdot l^i, 0)\ , \tag{1}$$

where $\rho(x,y)$ is the object albedo, and $n(x,y)$ is the normal to the object surface, given as

$$n(x,y) = \frac{(-z_x(x,y), -z_y(x,y), 1)}{\sqrt{1+\|\nabla z(x,y)\|_2^2}}. \tag{2}$$

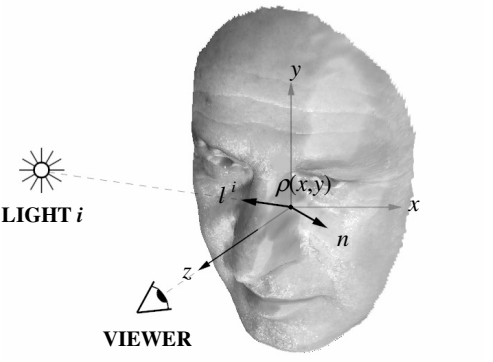



**Fig. 1.** 3D surface acquisition using photometric stereo

Using matrix-vector notation, Eq. (2) can be rewritten as

$$I(x,y) = \max(Lv, 0), \tag{3}$$

where

$$L = \begin{bmatrix} l_1^1 & l_2^1 & l_3^1 \\ \vdots & \vdots & \vdots \\ l_1^N & l_2^N & l_3^N \end{bmatrix}; \quad I(x,y) = \begin{bmatrix} I^1(x,y) \\ \vdots \\ I^N(x,y) \end{bmatrix}, \tag{4}$$

and

$$v_1 = -z_x v_3; \quad v_2 = -z_y v_3; \quad v_3 = \frac{\rho(x,y)}{\sqrt{1+\|\nabla z\|_2^2}}\ . \tag{5}$$

Given at least 3 linearly independent illuminations $\{l^i\}_{i=1}^N$ and the corresponding observations $\{I^i\}_{i=1}^N$, one can reconstruct the values of $\nabla z$ by pointwise least-squares solution

$$v = L^\dagger I(x,y)\ , \tag{6}$$

where $L^\dagger = (L^T L)^{-1} L^T$ denotes the Moore-Penrose pseudoinverse of $L$. When needed, the surface can be reconstructed by solving the Poisson equation

$$\tilde{z}_{xx} + \tilde{z}_{yy} = z_{xx} + z_{yy} \ , \tag{7}$$

with respect to $\tilde{z}$, which is the minimizer of the integral measure

$$\int\int \left( (\tilde{z}_x - z_x)^2 + (\tilde{z}_y - z_y)^2 \right) dxdy.$$

Photometric stereo is a simple 3D imaging method, which does not require expensive dedicated hardware. The assumption of Lambertian reflection holds for most parts of the human face (except the hair and the eyes) and makes this method very attractive for 3D face recognition application.

### 2.2 Structured Light

Proesmans et al. [11] and Winkelbach and Wahl [12] proposed a shape from 2D edge gradients reconstruction technique, which allows to reconstruct the surface normals (gradients) from two stripe patterns projected onto the object. The reconstruction technique is based on the fact that directions of the projected stripes in the captured 2D images depend on the local orientation of the surface in 3D. Classical edge-detecting operators can be used to find the direction of the stripe edges.

Figure 2 describes the relation between the surface gradient and the local stripe direction. A pixel in the image plane defines the viewing vector $s$. The stripe direction determines the stripe direction vector $v'$, lying in both the image plane and in the viewing plane. The real tangential vector of the projected stripe $v_1$ is perpendicular to the normal $c = v' \times s$ of the viewing plane and to the normal $p$ of the stripe projection plane. Assuming parallel projection, we obtain

$$v_1 = c \times p \ . \tag{8}$$

Acquiring a second image of the scene with a rotated stripe illumination relative to the first one, allows to calculate a second tangential vector $v_2$. Next, the surface normal is computed according to

$$n = v_1 \times v_2 \ . \tag{9}$$

In [13], Winkelbach and Wahl propose to use a single lighting pattern to estimate the surface normal from the local directions and *widths* of the projected stripes.

## 3 Bending-Invariant Representation

Human face can not be considered as a rigid object since it undergoes deformations resulting from facial expressions. On the other hand, the class of transformations that a facial surface can undergo is not arbitrary, and a suitable model

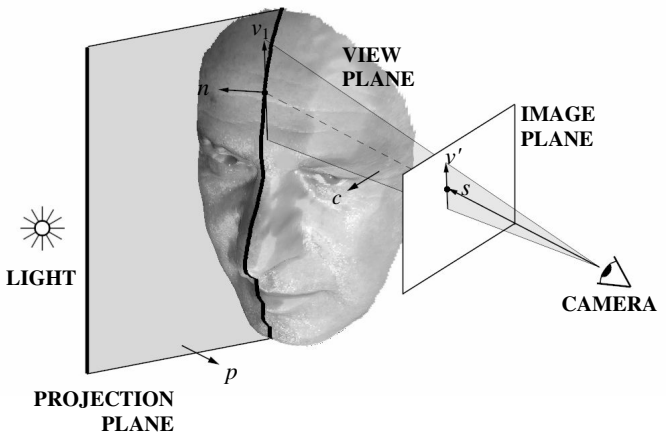



**Fig. 2.** 3D surface acquisition using structured light

for facial expressions is of *isometric* (or length-preserving) transformations [5]. Such transformations do not stretch or tear the surface, or more rigorously, preserve the surface metric. In face recognition application, faces can be thought of as an equivalence classes of surfaces obtained by isometric transformations. Unfortunately, classical surface matching methods, based on finding an Euclidean transformation of two surfaces which maximizes some shape similarity criterion (see, for example, [15], [16], [17]) usually fail to find similarities between two isometrically-deformed objects.

In [6], Elad and Kimmel introduced a deformable surface matching method, referred to as *bending-invariant canonical forms*, which was adopted in [5] for 3D face recognition. The key idea of this method is computation of invariant representations of the deformable surfaces, and then application of a rigid surface matching algorithm on the obtained invariants. We give a brief description of the method, necessary for the elaboration in Section 4.

Given a polyhedral approximation of the facial surface, $S$. One can think of such an approximation as if obtained by sampling the underlying continuous surface with a finite set of points $\{p_i\}_{i=1}^n$, and discretizing the metric associated with the surface

$$\delta(p_i, p_j) = \delta_{ij} \ . \tag{10}$$

We define the matrix of *squared mutual distances*,

$$(\Delta)_{ij} = \delta_{ij}^2 \ . \tag{11}$$

The matrix $\Delta$ is invariant under isometric surface deformations, but is not a unique representation of isometric surfaces, since it depends on arbitrary ordering and the selection of the surface points. We would like to obtain a geometric invariant, which would be unique for isometric surfaces on one hand, and will allow using simple rigid surface matching algorithms to compare such invariants on the

other. Treating the squared mutual distances as a particular case of dissimilarities, one can apply a dimensionality-reduction technique called multidimensional scaling (MDS) in order to embed the surface points with their geodesic distances in a low-dimensional Euclidean space $\mathbb{R}^m$ [10], [14], [6].

In [5] a particular MDS algorithm, the *classical scaling*, was used. The embedding into $\mathbb{R}^m$ is performed by first double-centering the matrix $\Delta$

$$B = -\frac{1}{n} J \Delta J \tag{12}$$

(here $J = I - \frac{1}{n}U$; $I$ is a $n \times n$ identity matrix, and $U$ is a matrix of ones). Then, the first $m$ eigenvectors $e_i$, corresponding to the $m$ largest eigenvalues of $B$, are used as the embedding coordinates

$$x_i^j = e_i^j; \qquad i = 1, ...n; \quad j = 1, ..., m \; . \tag{13}$$

where $x_i^j$ denotes the $j$-th coordinate of the vector $x_i$. The set of points $x_i$ obtained by the MDS is referred to as the bending-invariant canonical form of the surface; when $m = 3$, it can be plotted as a surface. Standard rigid surface matching methods can be used in order to compare between two deformable surfaces, using their bending-invariant representations instead of the surfaces themselves. Since the canonical form is computed up to a translation, rotation, and reflection transformation, in order to allow comparison between canonical forms, they must be aligned. This can be done by setting the first-order moments (center of mass) and the mixed second-order moments of the canonical form to zero (see [18]).

## 4 Measuring Geodesic Distances

One of the crucial steps in the construction of the canonical form of a given surface, is an efficient algorithm for the computation of geodesic distances on surfaces, that is, $\delta_{ij}$. A numerically consistent algorithm for distance computation on triangulated domains, henceforth referred to as Fast Marching on Triangulated Domains (FMTD), was used by Elad and Kimmel [6]. The FMTD was proposed by Kimmel and Sethian [7] as a generalization of the fast marching method [8]. Using FMTD, the geodesic distances between a surface vertex and the rest of the $n$ surface vertices can be computed in $O(n)$ operations. Measuring distances on manifolds was later done for graphs of functions [19] and implicit manifolds [20].

Since the main focus of this paper is how to *avoid* the surface reconstruction, we present a modified version of FMTD, which computes the geodesic distances on a surface, using the values of the surface gradient $\nabla z$ only. These values can be obtained, for example, from photometric stereo or structured light.

The facial surface can be thought of as a parametric manifold, represented by a mapping $X : \mathbb{R}^2 \to \mathbb{R}^3$ from the parameterization plane $U = (u^1, u^2) = (x, y)$ to the manifold

$$X(U) = (x^1(u^1, u^2), x^2(u^1, u^2), x^3(u^1, u^2)) \; ; \tag{14}$$

which, in turn, can be written as

$$X(U) = (x, y, z(x, y)) \ . \tag{15}$$

The derivatives of $X$ with respect to $u^i$ are defined as $X_i = \frac{\partial}{\partial u^i} X$, and they constitute a non-orthogonal coordinate system on the manifold (Figure 3). In the particular case of Eq. (15),

$$X_1(U) = (1, 0, z_x(x, y)); \qquad X_2(U) = (0, 1, z_y(x, y)) \ . \tag{16}$$

The distance element on the manifold is

$$ds = \sqrt{g_{ij} u^i u^j} \ , \tag{17}$$

where we use Einstein's summation convention, and the metric tensor $g_{ij}$ of the manifold is given by

$$(g_{ij}) = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} = \begin{bmatrix} X_1 \cdot X_1 & X_1 \cdot X_2 \\ X_2 \cdot X_1 & X_2 \cdot X_2 \end{bmatrix} \tag{18}$$

The classical Fast Marching method [8] calculates distances in an orthogonal coordinate system. The numerical stencil for the update of a grid point consists of the vertices of a right triangle. In our case, $g_{12} \neq 0$ and the resulting triangles are not necessarily right ones. If a grid point is updated by a stencil which is an obtuse triangle, a problem may arise. The values of one of the points of the stencil might not be set in time and cannot be used. There is a similar obstacle in Fast Marching on triangulated domains which include obtuse triangles [7].

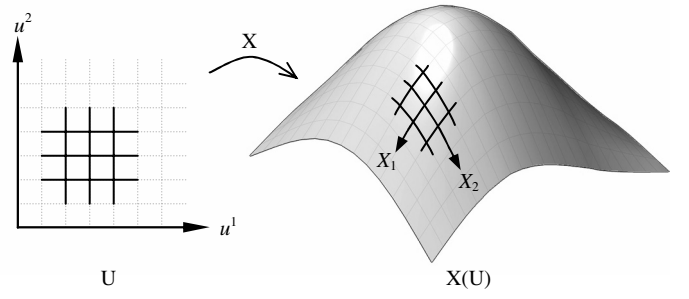



**Fig. 3.** The orthogonal grid on the parameterization plane $U$ is transformed into a non-orthogonal one on the manifold $X(U)$

Our solution is similar to that of [7]. We perform a preprocessing stage for the grid, in which we split every obtuse triangle into two acute ones (see Figure 4). The split is performed by adding an additional edge, connecting the updated grid point with a non-neighboring grid point. The distant grid point becomes

part of the numerical stencil. The need for splitting is determined according to the angle between the non-orthogonal axes at the grid point. It is calculated by

$$\cos\alpha = \frac{X_1 \cdot X_2}{\|X_1\|\|X_2\|} = \frac{g_{12}}{\sqrt{g_{11}g_{22}}} \,. \tag{19}$$

If $\cos\alpha = 0$ , the axes are perpendicular, and no splitting is required. If $\cos\alpha < 0$, the angle $\alpha$ is obtuse and should be split. The denominator in the rhs of Eq. (19) is always positive, so we need only check the sign of the numerator $g_{12}$. In order to split an angle, we should connect the updated grid point with another point, located $m$ grid points from the point in the $X_1$ direction, and $n$ grid points in the $X_2$ direction ($m$ and $n$ can be negative). The point is a proper supporting point, if the obtuse angle is split into two acute ones. For $\cos\alpha < 0$ this is the case if

$$\cos\beta_1 = \frac{X_1 \cdot (mX_1 + nX_2)}{\|X_1\|\|mX_1 + nX_2\|} = \frac{mg_{11} + ng_{12}}{\sqrt{g_{11}(m^2g_{11} + 2mng_{12} + n^2g_{22})}} > 0, \tag{20}$$

and

$$\cos\beta_2 = \frac{X_2 \cdot (mX_1 + nX_2)}{\|X_2\|\|mX_1 + nX_2\|} = \frac{mg_{12} + ng_{22}}{\sqrt{g_{22}(m^2g_{11} + 2mng_{12} + n^2g_{22})}} > 0 \,. \tag{21}$$

Also here, it is enough to check the sign of the numerators. For $\cos\alpha > 0$, $\cos\beta_2$ changes its sign and the constraints are

$$mg_{11} + ng_{12} > 0; \quad \text{and} \quad mg_{12} + ng_{22} < 0 \,. \tag{22}$$

This process is done for all grid points. Once the preprocessing stage is done, we have a suitable numerical stencil for each grid point and we can calculate the distances.

The numerical scheme used is similar to that of [7], with the exception that there is no need to perform the unfolding step. The supporting grid points that split the obtuse angles can be found more efficiently. The required triangle edge lengths and angles are calculated according to the surface metric $g_{ij}$ at the grid point, which, in turn, is computed using the surface gradients $z_x$, $z_y$. A more detailed description appears in [22].

## 5 3D Face Recognition Using Photometric Stereo without Surface Reconstruction

The modified FMTD method allows us to bypass the surface reconstruction stage in the 3D face recognition algorithm introduced in [5]. Instead, the values of the facial surface gradient $\nabla z$ is computed on a uniform grid using one of the methods discussed in Section 2 (see Figure 5). At the second stage, the raw data are preprocessed as proposed in [5]; in that paper, the preprocessing stage was limited to detecting the facial contour and cropping the parts of the face outside the contour.

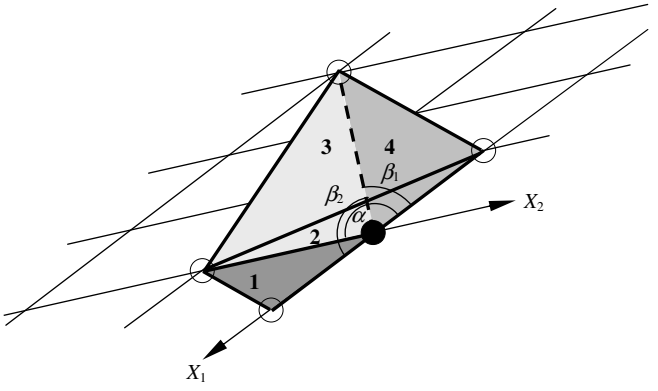



**Fig. 4.** The numerical support for the non-orthogonal coordinate system. Triangle 1 gives a proper numerical support, yet triangle 2 is obtuse. It is replaced by triangle 3 and triangle 4

**Fig. 5.** Surface gradient field (left), reconstructed surface (center) and its bending-invariant canonical form represented as a surface (right)

Next, an $n \times n$ matrix of squared geodesic distances is created by applying the modified FMTD from each of the $n$ selected vertices of the grid. Then, MDS is applied to the distance matrix, producing a canonical form of the face in a low-dimensional Euclidean space (three-dimensional in all our experiments). The obtained canonical forms are compared using a rigid surface matching algorithm. Texture is not treated in this paper.

As in [5], the method of moments described in [18] was used for rigid surface matching. The $(p, q, r)$-th moment of a three-dimensional surface is given by

$$M_{pqr} = \sum_n (x_n^1)^p (x_n^2)^q (x_n^3)^r \ , \tag{23}$$

where $x_n^i$ denotes the $i$-th coordinate of the $n$-th point in the surface samples. In order to compare between two surfaces, the vector of first $M$ moments $(M_{p_1 q_1 r_1}, ..., M_{p_M q_M r_M})$, termed as the *moment signature*, is computed for each

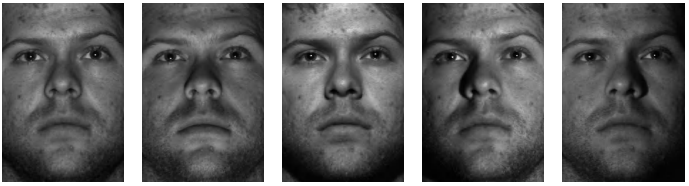



**Fig. 6.** A face from Yale Database B, acquired with different illuminations. Numbers in brackets indicate the azimuth and the elevation angle, respectively, determining the illumination direction

signature surface. The Euclidean distance between two moment signatures measures the dissimilarity between the two surfaces.

### 5.1 Experimental Results

In order to exemplify our approach, we performed an experiment, which demonstrates that comparison of canonical forms obtained without actual facial surface reconstruction, is in some cases, better than reconstruction and direct (rigid) comparison of the surfaces. It must be stressed that the purpose of the example is not to validate the 3D face recognition accuracy (which has been previously performed in [5]), but rather to test the feasibility of the proposed modified FMTD algorithm together with photometric stereo.

The Yale Face Database B [21] was used for the experiment. The database consisted of high-resolution grayscale images of different instances of 10 subjects of both Caucasian and Asian type, taken in controlled illumination conditions (Figure 6). Some instances of 7 subjects were taken from the database for the experiment. Direct surface matching consisted of the retrieval of the surface gradient according to Eq. (6) using 5 different illumination directions, reconstruction of the surface according to Eq. (7), alignment and computation of the surface moments signature according to Eq. (23). Canonical forms were computed from the surface gradient, aligned and converted into a moment signature according to Eq. (23).

In order to get some notion of the algorithms accuracy, we converted the relative distances between the subjects produced by each algorithm into 3D proximity patterns (Figure 7). These patterns, representing each subject as a point in $\mathbb{R}^3$, were obtained by applying MDS to the relative distances (with a distortion of less than 1%). The entire cloud of dots was partitioned into clusters formed by instances of the subjects $C_1$–$C_7$. Visually, the more $C_i$ are compact and distant from other clusters, the more accurate is the algorithm. Quantitatively, we measured (i) the variance $\sigma_i$ of $C_i$ and (ii) the distance $d_i$ between the centroid of $C_i$ and the centroid of the nearest cluster. Table 1 shows a quantitative comparison of the algorithms. Inter-cluster distances $d_i$ are given in units of the variance $\sigma_i$. Clusters $C_5$–$C_7$, consisting of a single instance of the

subject are not presented in the table. The use of canonical forms improved the cluster variance and the inter-cluster distance by about one order of magnitude, compared to direct facial surface matching.

**Table 1.** Properties of face clusters in Yale Database B using direct surface matching (dir) and canonical forms (can). $\sigma$ is the variance of the cluster and $d$ is the distance to the nearest cluster.

| Cluster | $\sigma_{dir}$ | $d_{dir}$ | $\sigma_{can}$ | $d_{can}$ |
|---|---|---|---|---|
| $C_1$ | 0.1749 | 0.1704 | 0.0140 | 4.3714 |
| $C_2$ | 0.2828 | 0.3745 | 0.0120 | 5.1000 |
| $C_3$ | 0.0695 | 0.8676 | 0.0269 | 2.3569 |
| $C_4$ | 0.0764 | 0.7814 | 0.0139 | 4.5611 |

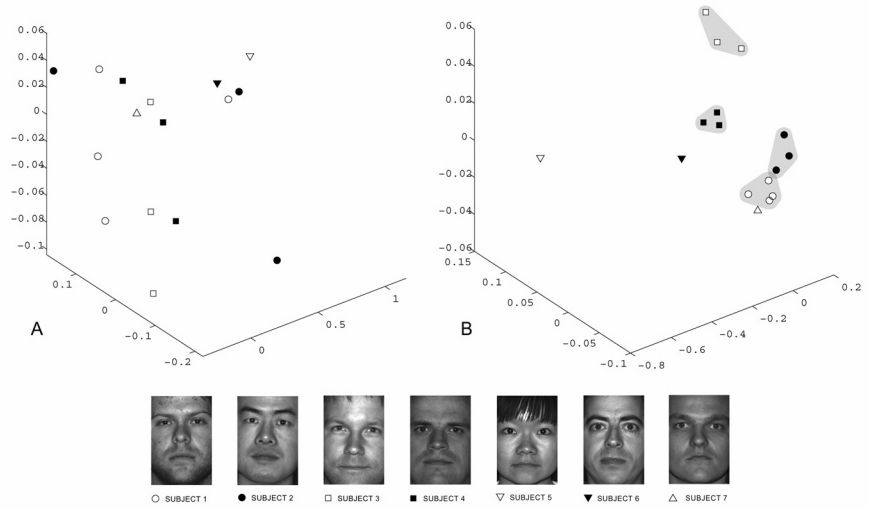



**Fig. 7.** Visualization of the face recognition results as three-dimensional proximity patterns. Subjects from the face database represented as points obtained by applying MDS to the relative distances between subjects. Shown here: straightforward surface matching (A) and canonical forms (B)

## 6 Conclusions

We have shown how to perform face recognition according to [5], without reconstructing the 3D facial surface. We used a modification of the Kimmel-Sethian

FMTD algorithm for computation of geodesic distances between points on the facial surface using only the surface metric tensor at each point. Our approach allows to use simple and efficient 3D acquisition techniques like photometric stereo for fast and accurate face recognition. Experimental results demonstrate feasibility of our approach for the task of face recognition.

**Acknowledgement.** This research was supported by Dvorah Fund of the Technion, Bar Nir Bergreen Software Technology Center of Excellence and the Technion V.P.R. Fund - E. and J. Bishop Research Fund.

## References

1. Gordon, G.: Face recognition from frontal and profile views. Proc. Int'l Workshop on Face and Gesture Recognition (1997) 47–52
2. Beumier, C., Acheroy, M. P.: Automatic face authentication from 3D surface. In: Proc. British Machine Vision Conf. (1988) 449–458
3. Huang, J., Blanz, V., Heisele, V.: Face recognition using component-based SVM classification and morphable models. In: SVM (2002) 334–341
4. Mavridis, N., Tsalakanidou, F., Pantazis, D., Malassiotis S., Strintzis, M. G.: The HISCORE face recognition application: Affordable desktop face recognition based on a novel 3D camera. In: Proc. Int'l Conf. Augmented Virtual Environments and 3D Imaging, Mykonos, Greece (2001)
5. Bronstein, A., Bronstein, M., Kimmel, R.: Expression-invariant 3D face recognition. In: Kittler, J., Nixon, M. (eds.): Proc. Audio and Video-based Biometric Person Authentication. Lecture Notes in Computer Science, Vol. 2688. Springer-Verlag, Berlin Heidelberg New York (2003) 62–69
6. Elad, A., Kimmel, R.: Bending invariant representations for surfaces. In: Proc. CVPR (2001) 168–174 (1997) 415–438
7. Kimmel, R., Sethian, J. A.: Computing geodesic on manifolds. In: Proc. US National Academy of Science Vol. 95 (1998) 8431–8435
8. Sethian, J. A.: A review of the theory, algorithms, and applications of level set method for propagating surfaces. In: Acta numerica (1996) 309–395
9. Bronstein, A., Bronstein, M., Gordon, E., Kimmel, R.: High-resolution structured light range scanner with automatic calibration. Techn. Report CIS-2003-06, Dept. Computer Science, Technion, Israel (2003)
10. Borg, I., Groenen, P.: Modern multidimensional scaling - theory and applications. Springer-Verlag, Berlin Heidelberg New York (1997)
11. Proesmans, M., Van Gool, L., Oosterlinck, A.: One-shot active shape acquisition. In: Proc. Internat. Conf. Pattern Recognition, Vienna, Vol. C (1996) 336–340
12. Winkelbach, S., Wahl, F. M.: Shape from 2D edge gradients. In: Lecture Notes in Computer Science, Vol. 2191. Springer-Verlag, Berlin Heidelberg New York (2001) 377–384
13. Winkelbach, S., Wahl, F. M.: Shape from single stripe pattern illumination. In: Van Gool, L. (ed.): Pattern Recognition DAGM. Lecture Notes in Computer Science, Vol. 2449. Springer-Verlag, Berlin Heidelberg New York (2002) 240–247
14. Schwartz, E. L., Shaw, A., Wolfson, E.: A numerical solution to the generalized mapmaker's problem: flattening nonconvex polyhedral surfaces. In: IEEE Trans. PAMI, Vol. 11 (1989) 1005–1008

15. Faugeras, O. D., Hebert, M.: A 3D recognition and positioning algorithm using geometrical matching between primitive surfaces. In: Proc. 7th Int'l Joint Conf. on Artificial Intelligence (1983) 996–1002
16. Besl, P. J.: The free form matching problem. In: Freeman, H. (ed.): Machine vision for three-dimensional scene. New York Academic (1990).
17. Barequet, G., Sharir, M.: Recovering the position and orientation of free-form objects from image contours using 3D distance map. In: IEEE Trans. PAMI, Vol. 19(9) (1997) 929–948
18. Tal, A., Elad, M., Ar, S.: Content based retrieval of VRML objects - an iterative and interactive approach. In: EG Multimedia Vol. 97 (2001) 97–108
19. Sethian, J., Vladimirsky, A.: Ordered upwind methods for static Hamilton-Jacobi equations: theory and applications. Techn. Report PAM 792, Center for Pure and Applied Mathematics, Univ. Calif. Berkeley (2001)
20. Memoli, F., Sapiro, G.: Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. In: Journal of Computational Physics, Vol. 173(2) (2001) 730–764
21. Yale Face Database B. http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html
22. Spira, A., Kimmel, R.: An efficient solution to the eikonal equation on parametric manifolds, In: INTERPHASE 2003 meeting, Isaac Newton Institute for Mathematical Sciences, 2003 Preprints, Preprint No. NI03045-CPD, UK (2003)

# Image and Video Segmentation by Anisotropic Kernel Mean Shift

Jue Wang, Bo Thiesson, Yingqing Xu, and Michael Cohen

Microsoft Research (Asia and Redmond)

**Abstract.** Mean shift is a nonparametric estimator of density which has been applied to image and video segmentation. Traditional mean shift based segmentation uses a radially symmetric kernel to estimate local density, which is not optimal in view of the often structured nature of image and more particularly video data. In this paper we present an anisotropic kernel mean shift in which the shape, scale, and orientation of the kernels adapt to the local structure of the image or video. We decompose the anisotropic kernel to provide handles for modifying the segmentation based on simple heuristics. Experimental results show that the anisotropic kernel mean shift outperforms the original mean shift on image and video segmentation in the following aspects: 1) it gets better results on general images and video in a smoothness sense; 2) the segmented results are more consistent with human visual saliency; 3) the algorithm is robust to initial parameters.

## 1 Introduction

Image segmentation refers to identifying homogenous regions in the image. Video segmentation, in this paper, means the joint spatial and temporal analysis on video sequences to extract regions in the dynamic scenes. Both of these tasks are misleadingly difficult and have been extensively studied for several decades. Refer to [9,10,11] for some good surveys. Generally, spatio-temporal video segmentation can be viewed as an extension of image segmentation from a 2D to a 3D lattice. Recently, mean shift based image and video segmentation has gained considerable attention due to its promising performance.

Many other data clustering methods have been described in the literature, ranging from top down methods such as K-D trees, to bottom up methods such as K-means and more general statistical methods such as mixtures of Gaussians. In general these methods have not performed satisfactorily for image data due to their reliance on an a priori parametric structure of the data segment, and/or estimates of the number of segments expected. Mean shift's appeal is derived from both its performance and its relative freedom from specifying an expected number of segments. As we will see, this freedom has come at the cost of having to specify the size (bandwidth) and shape of the influence kernel for each pixel in advance.

The difficulty in selecting the kernel was recognized in [3,4,12] and was addressed by automatically determining a bandwidth for spherical kernels. These

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 238–249, 2004.
© Springer-Verlag Berlin Heidelberg 2004

approaches are all purely *data driven*. We will leverage this work and extend it to automatically select general elliptical (anisotropic) kernels for each pixel. We also add a priori knowledge about typical structures found in video data to take advantage of the extra freedom in the kernels to adapt to the local structure.

### 1.1 Mean Shift Based Image and Video Segmentation

Rather than begin from an initial guess at the segmentation, such as seeding points in the K-means algorithm, mean shift begins at each data point (or pixel in an image or video) and first estimates the local density of similar pixels (i.e., the density of nearby pixels with similar color). As we will see, carefully defining "nearby" and "similar" can have an important impact on the results. This is the role the kernel plays.

More specifically, mean shift algorithms estimate the local density *gradient* of similar pixels. These gradient estimates are used within an iterative procedure to find the peaks in the local density. All pixels that are drawn upwards to the same peak are then considered to be members of the same segment.

As a general nonparametric density estimator, mean shift is an old pattern recognition procedure proposed by Fukunage and Hostetler [7], and its efficacy on low-level vision tasks such as segmentation and tracking has been extensively exploited recently. In [1,5], it was applied for continuity preserving filtering and image segmentation. Its properties were reviewed and its convergence on lattices was proven. In [2], it was used for non-rigid objects tracking and a sufficient convergence condition was given. Applying mean shift on a 3D lattice to get a spatio-temporal segmentation of video was achieved in [6], in which a hierarchical strategy was employed to cluster pixels of 3D space-time video stack, which were mapped to 7D feature points (position(2), time(1), color(3), and motion(1)).

The application of mean shift to an image or video consists of two stages. The first stage is to define a *kernel* of influence for each pixel $x_i$. This kernel defines a measure of intuitive *distance* between pixels, where distance encompasses both spatial (and temporal in the case of video) as well as color distance. Although manual selection of the size (or *bandwidth*) and shape of the kernel can produce satisfactory results on general image segmentation, it has a significant limitation. When local characteristics of the data differ significantly across the domain, it is difficult to select globally optimal bandwidths. As a result, in a segmented image some objects may appear too coarse while others are too fine. Some efforts have been reported to locally vary the bandwidth. Singh and Ahuja [12] determine local bandwidths using Parzen windows to mimic local density. Another variable bandwidth procedure was proposed in [3] in which the bandwidth was enlarged in sparse regions to overcome the noise inherent with limited data.

Although the size may vary locally, all the approaches described above used a radially symmetric kernel. One exception is the recent work in [4] that describes the possibility of using the general local covariance to define an asymmetric kernel. However, this work goes on to state, "Although a fully parameterized covariance matrix can be computed.., this is not necessarily advantageous.." and then returns to the use of radially symmetric kernels for reported results.

The second iterative stage of the mean shift procedure assigns to each pixel a *mean shift point*, $M(x_i)$, initialized to coincide with the pixel. These mean shift points are then iteratively moved upwards along the gradient of the density function defined by the sum of all the kernels until they reach a stationary point (a *mode* or hilltop on the virtual terrain defined by the kernels). The pixels associated with the set of mean shift points that migrate to the (approximately) same stationary point are considered to be members of a single segment. Neighboring segments may then be combined in a post process.

Mathematically, the general multivariate kernel density estimate at the point, $x$, is defined by

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K_H(x - x_i) \tag{1}$$

where the $n$ data points $x_i$ represent a sample from some unknown density $f$, or in the case of images or video, the pixels themselves.

$$K_H(x) = |H|^{-1/2} K(H^{-1/2} x) \tag{2}$$

where $K(z)$ is the $d$-variate kernel function with compact support satisfying the regularity constraints as described in [13], and $H$ is a symmetric positive definite $d \times d$ bandwidth matrix. For the radially symmetric kernel, we have

$$K(z) = c\, k(||z||^2) \tag{3}$$

where $c$ is the normalization constant. If one assumes a single global spherical bandwidth, $H = h^2 I$, the kernel density estimator becomes

$$\hat{f}(x) = \frac{1}{n(h)^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{4}$$

For image and video segmentation, the feature space is composed of two independent domains: the *spatial/lattice* domain and the *range/color* domain. We map a pixel to a multi-dimensional feature point which includes the $p$ dimensional spatial lattice ($p = 2$ for image and $p = 3$ for video) and $q$ dimensional color ($q = 3$ for L*u*v color space). Due to the different natures of the domains, the kernel is usually broken into the product of two different radially symmetric kernels (superscript $s$ will refer to the spatial domain, and $r$ to the color range):

$$K_{h^s,h^r}(x) = \frac{c}{(h^s)^p (h^r)^q} k^s \left( \left\| \frac{x^s}{h^s} \right\|^2 \right) k^r \left( \left\| \frac{x^r}{h^r} \right\|^2 \right) \tag{5}$$

where $x^s$ and $x^r$ are respectively the spatial and range parts of a feature vector, $k^s$ and $k^r$ are the profiles used in the two domains, $h^s$ and $h^r$ are employed bandwidths in two domains, and $c$ is the normalization constant. With the kernel from (5), the kernel density estimator is

$$\hat{f}(x) = \frac{c}{n(h^s)^p (h^r)^q} \sum_{i=1}^{n} k^s \left( \left\| \frac{x^s - x_i^s}{h^s} \right\|^2 \right) k^r \left( \left\| \frac{x^r - x_i^r}{h^r} \right\|^2 \right) \tag{6}$$

As apparent in Equations (5) and (6), there are two main parameters that have to be defined by the user for the simple radially symmetric kernel based approach: the spatial bandwidth $h^s$ and the range bandwidth $h^r$. In the variable bandwidth mean shift procedure proposed in [3], the estimator (6) is changed to

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{c}{(h_i^s)^p\ (h_i^r)^q} k^s \left( \left\| \frac{x^s - x_i^s}{h_i^s} \right\|^2 \right) k^r \left( \left\| \frac{x^r - x_i^r}{h_i^r} \right\|^2 \right) \tag{7}$$

There are now important differences between (6) and (7). First, potentially different bandwidths $h_i^s$ and $h_i^r$ are assigned to each pixel, $x_i$, as indicated by the subscript $i$. Second, the different bandwidths associated with each point appear within the summation. This is the so-called sample point estimator [3], as opposed to the *balloon* estimator defined in Equation (6). The sample point estimator, which we will refer to as we proceed, ensures that all pixels respond to the same global density estimation during the segmentation procedure. Note that the sample point and balloon estimators are the same in the case of a single globally applied bandwidth.

### 1.2 Motivation for an Anisotropic Kernel

During the iterative stage of the mean shift procedure, the mean shift points associated with each pixel climb to the hilltops of the density function. At each iteration, each mean shift point is attracted in varying amounts by the sample point kernels centered at nearby pixels. More intuitively, a kernel represents a measure of the likelihood that other points are part of the same segment as the point under the kernel's center. With no a priori knowledge of the image or video, actual distance (in space, time, and color) seems an obvious (inverse) correlate for this likelihood; the closer two pixels are to one another the more likely they are to be in the same segment.

We can, however, take advantage of examining a local region surrounding each pixel to select the size and shape of the kernel. Unlike [3], we leverage the full local covariance matrix of the local data to create a kernel with a general elliptical shape. Such kernels adapt better to non-compact (i.e., long skinny) local features such as can be seen in the monkey bars detail in Figure 2 and the zebra stripes in Figure 5. Such features are even more prevalent in video data from stationary or from slowly or linearly moving cameras. When considering video data, a spatio-temporal slice (parallel to the temporal axis) is as representative of the underlying data as any single frame (orthogonal to the temporal axis). Such a slice of video data exhibits stripes with a slope relative to the speed at which objects move across the visual field (see Figures 3 and 4). The problems in the use of radially symmetric kernels is particularly apparent in these spatio-temporal slice segmentations. The irregular boundaries between and across the stripe-like features cause a lack of temporal coherence in the video segmentation.

An anisotropic kernel can adapt its profile to the local structure of the data. The use of such kernels proves more robust, and is less sensitive to initial parameters compared with symmetric kernels. Furthermore, the anisotropic kernel

provides a set of handles for application-driven segmentation. For instance, a user may desire that the still background regions be more coarsely segmented while the details of the moving objects to be preserved when segmenting a video sequence. To achieve this, we simply expand those local kernels (in the color and/or spatial dimensions) whose profiles have been elongated along the time dimension. By providing a set of heuristic rules described below on how to modulate the kernels, the segmentation strategy can be adapted to various applications.

## 2 Anisotropic Kernel Mean Shift

### 2.1 Definition

The *Anisotropic Kernel Mean Shift* associates with each data point (a pixel in an image or video) an anisotropic kernel. The kernel associated with a pixel adapts to the local structure by adjusting its shape, scale, and orientation. Formally, the density estimator is written as

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h^r(H_i^s)^q} k^s\left(g(x^s, x_i^s, H_i^s)\right) k^r\left(\left\|\frac{x^r - x_i^r}{h^r(H_i^s)}\right\|^2\right) \tag{8}$$

where $g(x^s, x_i^s, H_i^s)$ is the Mahalanobis metric in the spatial domain:

$$g(x^s, x_i^s, H_i^s) = (x_i^s - x^s)^T H_i^{s\,-1}(x_i^s - x^s) \tag{9}$$

In this paper we use a spatial kernel with a constant profile, $k^s(z) = 1$ if $|z| < 1$, and 0 otherwise. For the color domain we use an Epanechnikov kernel with a profile $k^r(z) = 1-|z|$ if $|z| < 1$ and 0 otherwise. Note that in our definition, the bandwidth in color range $h^r$ is a function of the bandwidth matrix in space domain $H_i^s$. Since $H_i^s$ is determined by the local structure of the video, $h^r$ thus varies from one pixel to another. Possibilities on how to modulate $h^r$ according to $H^s$ will be discussed later.

The bandwidth matrix $H_i^s$ is symmetric positive definite. If it is simplified into a diagonal matrix with equal diagonal elements, (i.e., a scaled identity), then $H_i^s$ models the radially symmetric kernels. In the case of video data, the time dimension may be scaled differently to represent notions of equivalent "distance" in time vs. image space. In general, allowing the diagonal terms to be scaled differently allows for the kernels to take on axis aligned ellipsoidal shapes. A full $H_i^s$ matrix provides the freedom to model kernels of a general ellipsoidal shape oriented in any direction. The Eigen vectors of $H_i^s$ will point along the axes of such ellipsoids. We use this additional freedom to shape the kernels to reflect local structures in the video as described in the next section.

### 2.2 Kernel Modulation Strategies

Anisotropic kernel mean shift give us a set of handles on modulating the kernels during the mean shift procedure. How to modulate the kernel is application related and there is not an uniform theory for guidance. We provide some intuitive

heuristics for video data with an eye towards visually salient segmentation. In the case of video data we want to give long skinny segments at least an equal chance to form as more compact shapes. These features often define the salient features in an image. In addition, they are often very prominent features in the spatio-temporal slices as can be seen in many spatio-temporal diagrams. In particular, we want to recognize segments with special properties in the time domain. For example, we may wish to allow static objects to form into larger segments while moving objects to be represented more finely with smaller segments.

An anisotropic bandwidth matrix $H_i^s$ is first estimated starting from a standard radially symmetric diagonal $H_i^s$ and color radius $h^r$. The neighborhood of pixels around $x_i$ is defined by those pixels whose position, $x$, satisfies

$$k^s(g(x, x_i, H_i^s)) < 1;\ k^r\left(\left\|\frac{x - x_i}{h^r(H_i^s)}\right\|^2\right) < 1 \tag{10}$$

An analysis of variance of the points within the neighborhood of $x_i$ provides a new full matrix $\bar{H}_i^s$ that better describes the local neighborhood of points.

To understand how to modulate the full bandwidth matrix $\bar{H}_i^s$, it is useful to decompose it as

$$\bar{H}_i^s = \lambda D A D^T \tag{11}$$

where $\lambda$ is a global scalar, $D$ is a matrix of normalized Eigen vectors, and $A$ is a diagonal matrix of Eigen values which is normalized to satisfy:

$$\prod_{i=1}^{p} a_i = 1 \tag{12}$$

where $a_i$ is the $i^{th}$ diagonal elements of $A$, and $a_i \geq a_j$, for $i < j$. Thus, $\lambda$ defines the overall volume of the new kernel, $A$ defines the relative lengths of the axes, and $D$ is a rotation matrix that orients the kernel in space and time.

We now have intuitive handles for modulating the anisotropic kernel. The $D$ matrix calculated by the covariance analysis is kept unchanged during the modulation process to maintain the orientation of the local data. By adjusting $A$ and $\lambda$, we can control the spatial size and shape of the kernel. For example, we can encourage the segmentation to find long skinny regions by diminishing the smaller Eigen values in $A$ as

$$a_i' = \begin{cases} a_i^{3/2} & : \quad a_i <= 1 \\ \sqrt{a_i} & : \quad a_i > 1 \end{cases}, i = 2, ..., p \tag{13}$$

In this way the spatial kernel will stretch more in the direction in which the object elongates. To create larger segments for static objects we detect kernels oriented along the time axis as follows. First, a scale factor $s_t$ is computed as

$$s_t = \alpha + (1 - \alpha) \prod_{i=1}^{p-1} d_1(i)^2 \tag{14}$$

where $d_1$ is the first Eigen vector in $D$, which corresponds with the largest Eigen value $a_1$. $d_1(i)$ stands for the $ith$ element in $d_1$, which is the x, y and t component of the vector when $i = 1, 2, 3$, respectively. $\alpha$ is a constant between 0 and 1. In our system, we set $\alpha$ to 0.25. The product in the above equation corresponds to the cosine of the angle between the first Eigen vector and the time axis. If the stretch direction of the kernel is close to the time axis, the scale factor is close to a small value $\alpha$. Otherwise if the stretch direction is orthogonal to the time axis, then $s_t$ is close to 1. The matrix $A$ is thus changed as

$$a'_i = a_i \cdot s_t, i = 2, ..., p \tag{15}$$

After the matrix $A$ is modified by (13) and/or (14), the global scalar $\lambda$ is changed correspondingly as

$$\lambda' = \lambda \prod_{i=1}^{p} \left( \frac{a_i}{a'_i} \right) \tag{16}$$

To keep the *analysis resolution* in the color domain consistent with that in space domain, the bandwidth in the color domain is changed to

$$h^r(H_i^s) \leftarrow \sqrt{\frac{\lambda'}{\lambda}} \cdot h^r(H_i^s) \tag{17}$$

The effect is to increase the color tolerance for segments that exhibit a large stretch, typically along the time axis (i.e., are static in the video).

### 2.3 Algorithm

The anisotropic mean shift segmentation is very similar to the traditional mean shift segmentation algorithm. The only difference is that a new anisotropic spatial kernel and space dependent kernel in the color domain are determined individually for each feature point prior to the main mean shift procedure. Recall that when kernels vary across feature points, the sample point estimator should be used in the mean shift procedure (note subscripts $j$ within summation in step 4. The sample point anisotropic mean shift algorithm is formally described below. Steps 1-3 are the construction of kernels and steps 4-6 is the main mean shift procedure for these kernels.

1. Data and kernel initialization.
   - Transfer pixels into multidimensional (5D for image, 6D for video) feature points, $x_i$.
   - Specify initial spatial domain parameter $h_0^s$ and initial range domain parameter $h_0^r$.
   - Associate kernels with feature points, initialize means to these points.
   - Set all initial bandwidth matrices in the spatial domain as the diagonal matrix $H_i^s = (h_0^s)^2 I$. Set all initial bandwidths in the range domain as $h^r(H_i^s) = h_0^r$.

2. For each point $x_i$, determine the anisotropic kernel and related color radius:
   - Search the neighbors of $x_i$ to get all the points $x_j$, $j = 1, ..., n$ that satisfy the constraints of kernels:
   $$k^s(g(x_i, x_j, H_i^s)) < 1; k^r\left(\left\|\frac{x_i - x_j}{h^r(H_i^s)}\right\|^2\right) < 1$$
   Update the bandwidth matrix $H_i^s$ as:
   $$H_i^s \leftarrow \frac{\Sigma_{j=1}^n \left\|\frac{x_i^r - x_j^r}{h^r(H_i^s)}\right\|^2 (x_j^s - x_i^s)(x_j^s - x_i^s)^T}{\Sigma_{j=1}^n \left\|\frac{x_i^r - x_j^r}{h^r(H_i^s)}\right\|^2}$$
   - Modulate $H_i^s$ as discussed in the previous section. For image segmentation, apply the modulations for exaggerating eccentricity (13) and modifying overall scale (16) sequentially; for video segmentation, sequentially apply the modulations for eccentricity (13), scaling for static segments (15), and overall scale (16).
   - Modulate color tolerance $h^r(H_i^s)$ as described in (17).
3. Repeat step (2) a fixed number of times (typically 3).
4. Associate a mean shift point $M(x_i)$ with every feature point (pixel), $x_i$, and initialize it to coincide with that point. Repeat for each $M(x_i)$
   - Determine the neighbors, $x_j$, of $M(x_i)$ as in (18) replacing $x_i$ with $M(x_i)$.
   - Calculate the mean shift vector summing over the neighbors:
   $$M_v(x_i) = \frac{\Sigma_{j=1}^n (x_j - M(x_i)) \left\|\frac{M(x_i^r) - x_j^r}{h^r(H_j^s)}\right\|^2}{\Sigma_{j=1}^n \left\|\frac{M(x_i^r) - x_j^r}{h^r(H_j^s)}\right\|^2}$$
   - Update the mean shift point:
   $$M(x_i) \leftarrow M(x_i) + M_v(x_i)$$
   until $M_v(x_i)$ is less than a specified epsilon.
5. Merge pixels whose mean vectors are approximately the same to produce homogenous color regions.
6. Optionally, eliminate segments containing less than a given number of pixels.

### 2.4 Initial Scale Selection

As in traditional mean shift image segmentation, the anisotropic kernel mean shift segmentation algorithm also relies on two initial parameters: the initial bandwidths in space and range domain. However, since the bandwidth matrices $H_i^s$ and the bandwidth in range domain $h^r(H_i^s)$ are adaptively modulated, the proposed algorithm is more robust to the initial parameters.

To further increase the robustness, one may also adopt the semiparametric scale selection method described in [3]. The system automatically determines an initial spatial bandwidth for each kernel associated with a point. The user is thus required to set only one parameter: the bandwidth $h_0^r$ in range domain. The local scale is given as the bandwidth that maximizes the norm of the normalized mean shift vector. Refer to [3] for the detailed description and proof.

## 3 Results

We have experimented with the anisotropic mean shift procedure outlined above on a number of video and still imagery. The first set of images are taken from a short 10 second video of a girl swinging on monkey bars taken from a stationary camera. We first examine a ten frame sequence. We segmented the frames in three ways: 1) each individually with a standard radially symmetric kernel, 2) segmenting the 3D block of video with radially symmetric kernels, and 3) with 3D anisotropic kernels. The results are shown in Figure 1 along with summed pairwise differences between frames. The expected temporal coherence from the stationary camera is faithfully captured in the anisotropic case. A detail of the monkey bars (Figure 2) shows how salient features such as the straight bars are also better preserved. Finally, we show the comparison of symmetric vs. anisotropic kernels on spatio-temporal slices from the monkey bars sequence (Figure 3) and the well known garden sequence (Figure 4) that show much improved segmentation along the trajectories of objects typically found in video. A last example run on a zebra image shows improvement as well in capturing long thin features.

### 3.1 Robustness

The anisotropic kernel mean shift is more robust to initial parameters than the traditional mean shift. To test this, we correlated the number of segmented regions to the *analysis resolution* on the monkey bars spatio-temporal slice. We fixed $h_r$ to be 6.5 (in the 0 to 255 color space) in both cases. The analysis resolution is then defined as $h^s$ for the fixed symmetric kernels, and the average $\lambda$ value from the decomposition of the $H_i^s$ in equation (11). As expected, the number of segments increases as the analysis resolution decreases in both cases (see Figure 2). However, the slope is almost twice as steep in the radially symmetric case as with the anisotropic kernel. This indicates that the traditional algorithm is more sensitive to initial parameters than the proposed algorithm. Furthermore, by incorporating the scale selection method, the algorithm automatically selects initial spatial bandwidth.

## 4 Discussion

Mean shift methods have gained popularity for image and video segmentation due to their lack of reliance on a priori knowledge of the number of expected segments. Most previous methods have relied on radially symmetric kernels. We have shown why such kernels are not optimal, especially for video that exhibits long thin structures in the spatio-temporal slices. We have extended mean shift to allow for anisotropic kernels and demonstrated their superior behavior on both still images and a short video sequence.

The anisotropic kernels plus the sample point density estimation both make the inner loop of the mean shift procedure more complex. We are currently

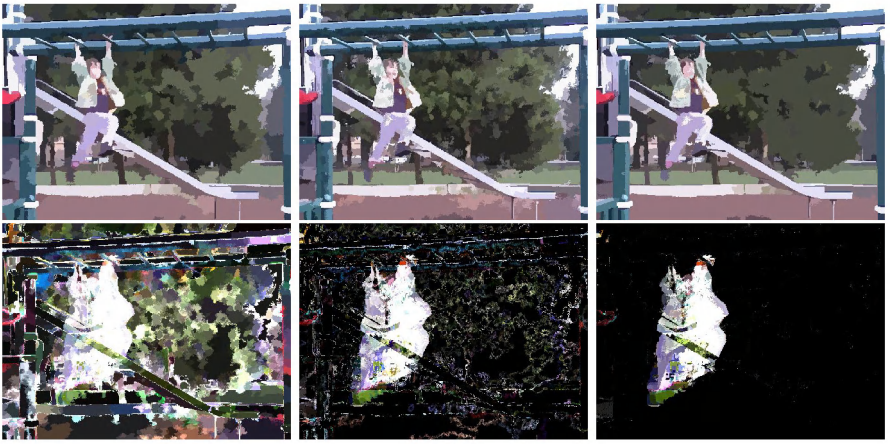


**Fig. 1.** First row: Segmentation for 2D radially symmetric kernel, 3D symmetric kernel, 3D anisotropic kernel. Note the larger background segments in the anisotropic case while preserving detail in the girl. Second row: total absolute differences across nine pairs of subsequent frames in a ten frame sequence, 2D, 3D radially symmetric, 3D anisotropic. Note the clean segmentation of the moving girl from the background.

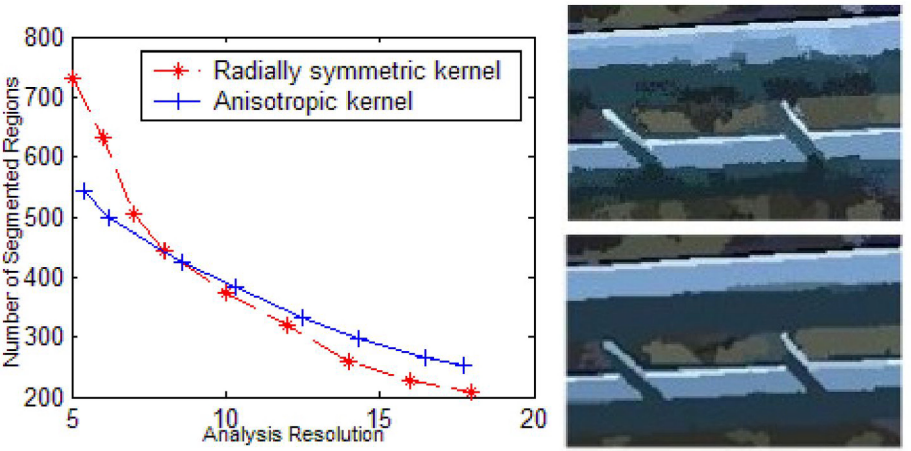



**Fig. 2.** Left: Robustness results. Right: Monkey Bar detail between 3D radially symmetric kernel result (top) and anisotropic result (bottom).

working on ways to make this more efficient by recognizing pixels that move together early in the iterative process.

It would be nice to have a formal way to objectively analyze the relative success of different mean shift segmentation procedures. Applications such as determining optical flow directly from the kernel orientations might provide a

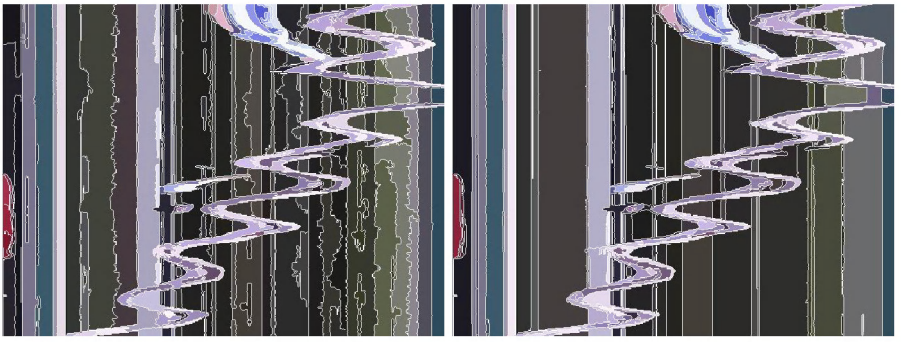


**Fig. 3.** Spatio-temporal slice of 10 second video segmented by radially symmetric kernel mean shift (left, 384 segments) and anisotropic kernel mean shift (right, 394 segments). Note the temporal coherence indicated by the straight vertical segmentation.

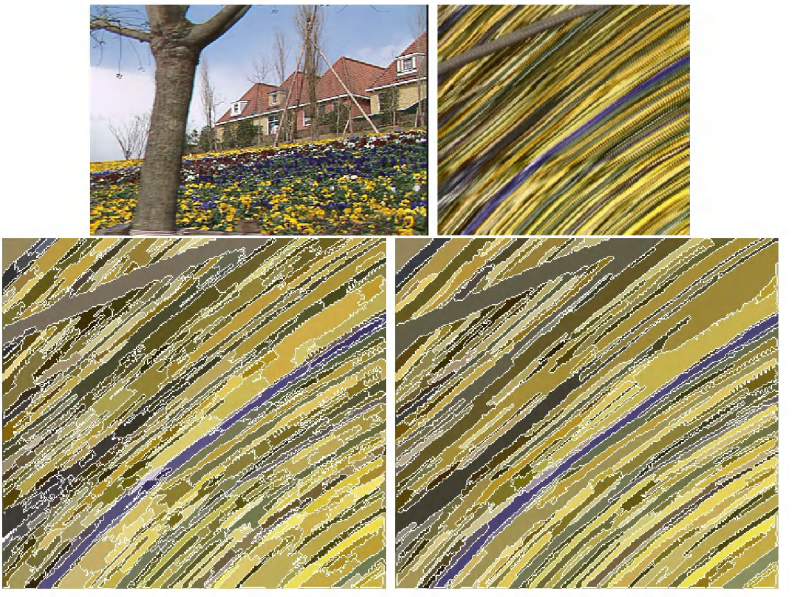


**Fig. 4.** Well known garden sequence frame and an epipolar slice. Radially symmetric and Anisotropic segmentation (267 and 266 segments).

useful metric. We also look forward to applying our methods to one of our original motivations; automatically producing cartoon-like animations from video.

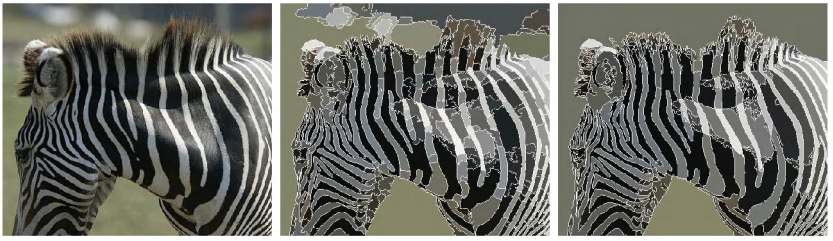


**Fig. 5.** Zebra photograph. Segmentation with radially symmetric and anisotropic kernels (386 and 387 segments).

## References


1. Comaniciu, D., Meer, P.: Mean shift analysis and applications. Proc. IEEE Int. Conf. on Computer Vision, Greece (1999) 1197-1203.
2. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (2000) 142-151.
3. DeMenthon, D., Megret, R.: The variable bandwidth mean shift and data-driven scale selection. Proc. IEEE 8th Int. Conf. on Computer Vision, Canada (2001) 438-445.
4. Comaniciu, D.: An Algorithm for Data-Driven Bandwidth Selection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 2, February 2003 (2003).
5. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. on PAMI (2002) 603-619.
6. DeMenthon, D., Megret, R.: Spatio-temporal segmentation of video by hierarchical mean shift analysis. Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition (2000) 142-151.
7. Fukunaga, K., Hostetler, L.: The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Information Theory **21** (1975) 32–40.
8. Lorensen, W.E., Cline, H.E.: Marching Cubes: a high resolution 3D surface reconstruction algorithm. Proc. ACM SIGGRAPH 1987, (1987) 163-169.
9. Megret, R., DeMenthon, D.: A Survey of Spatio-Temporal Grouping Techniques. Technical report: LAMP-TR-094/CS-TR-4403, University of Maryland, College Park (1994).
10. Pal, N.R., Pal, S.K.: A review on image segmentation techniques. Pattern Recognition **26** 9 (1993) 1277–1294.
11. Skarbek, W., Koschan, A.: Colour Image Segmentation: A survey. Technical report, Technical University Berlin (1994).
12. Singh, M., Ahuja, N.: Regression Based Bandwidth Selection for Segmentation using Parzen Windows. Proc. IEEE International Conference on Computer Vision (2003) **1** 2–9.
13. Wand, M., Jones, M.: Kernel Smoothing. Chapman & Hall (1995) p. 95.

# Colour Texture Segmentation by Region-Boundary Cooperation

Jordi Freixenet, Xavier Muñoz, Joan Martí, and Xavier Lladó

Computer Vision and Robotics Group. University of Girona
Campus de Montilivi 17071. Girona, Spain
{jordif,xmunoz,joanm,llado}@eia.udg.es

**Abstract.** A colour texture segmentation method which unifies region and boundary information is presented in this paper. The fusion of several approaches which integrate both information sources allows us to exploit the benefits of each one. We propose a segmentation method which uses a coarse detection of the perceptual (colour and texture) edges of the image to adequately place and initialise a set of active regions. Colour texture of regions is modelled by the conjunction of non-parametric techniques of kernel density estimation, which allow to estimate the colour behaviour, and classical co-occurrence matrix based texture features. When the region information is defined, accurate boundary information can be extracted. Afterwards, regions concurrently compete for the image pixels in order to segment the whole image taking both information sources into account. In contrast with other approaches, our method achieves relevant results on images with regions with the same texture and different colour (as well as with regions with the same colour and different texture), demonstrating the performance of our proposal. Furthermore, the method has been quantitatively evaluated and compared on a set of mosaic images, and results on real images are shown and analysed.

## 1 Introduction

Image segmentation has been, and still is, a relevant research area in Computer Vision, and hundreds of segmentation algorithms have been proposed in the last 30 years. Many segmentation methods are based on two basic properties of the pixels in relation to their local neighbourhood: discontinuity and similarity. Methods based on pixel discontinuity are called boundary-based methods, whereas methods based on pixel similarity are called region-based methods. However, it is well known that such segmentation techniques - based on boundary or region information alone - often fail to produce accurate segmentation results [1]. Hence, in the last few years, there has been a tendency towards algorithms which take advantage of the complementary nature of such information.

Reviewing the different works on region-based segmentation which have been proposed (see surveys on image segmentation [2,3]), it is interesting to note the evolution of region-based segmentation methods, which were initially focused on

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 250–261, 2004.
© Springer-Verlag Berlin Heidelberg 2004

grey-level images, and which gradually incorporated colour, and more recently, texture. In fact, colour and texture are fundamental features in defining visual perception and experiments have demonstrated that the inclusion of colour can increase the texture segmentation/classification results without significantly complicating the feature extraction algorithms [4]. Nevertheless, most of the literature deals with segmentation based on either colour or texture, and there is a limited number of systems which consider both properties together.

In this work we propose a new strategy for the segmentation of colour texture images. Having reviewed and analysed more than 50 region-boundary cooperative algorithms, we have clearly identified 7 different strategies (see [5]) to perform the integration. As a natural development of this review work, we defined a new strategy for image segmentation [6] based on a combination of different methods used to integrate region and boundary information. Moreover, to knowledge of the authors, there has not yet been any proposal which integrates region and boundary information sources while taking colour and texture properties into account. Hence, we have extended our previous approach to deal with the problem of colour texture segmentation. We focus on "color texture" taking into account that is both spatial and statistical. It is spatial since texture is the relationship of groups of pixels. Nothing can be learned about texture from an isolated pixel, and little from a histogram of pixel values.

The remainder of this paper is organised as follows: a review of the recent work on color texture segmentation concludes this introduction. Section 2 describes the proposed segmentation strategy detailing the placement of starting seeds, the definition of region and boundary information and the growing of active regions. The experimental results concerning a set of synthetic and real images demonstrating the validity of our proposal appear in Section 3. Finally, conclusions are given in Section 4.

### 1.1 Related Work

Most of the literature deals with segmentation based on either colour or texture. Although colour is an intrinsic attribute of an image and provides more information than a single intensity value, there has been few attempts to incorporate chrominance information into textural features [4]. This extension to colour texture segmentation was originated by the intuition that using information provided by both features, one should be able to obtain more robust and meaningful results.

A rather limited number of systems use combined information of colour and texture, and even when they do so, both aspects are mostly dealt with using separate methods [7]. Generally, two segmentations are computed for colour and texture features independently, and obtained segmentations are then merged into a single colour texture segmentation result with the aim of preserving the strength of each modality: smooth regions and accurate boundaries using texture and colour segmentation, respectively [8,9]. The main drawback is related to the selection rule for assigning the appropriate segmentation labels to the final segmentation result, where segmentation maps disagree with each other.

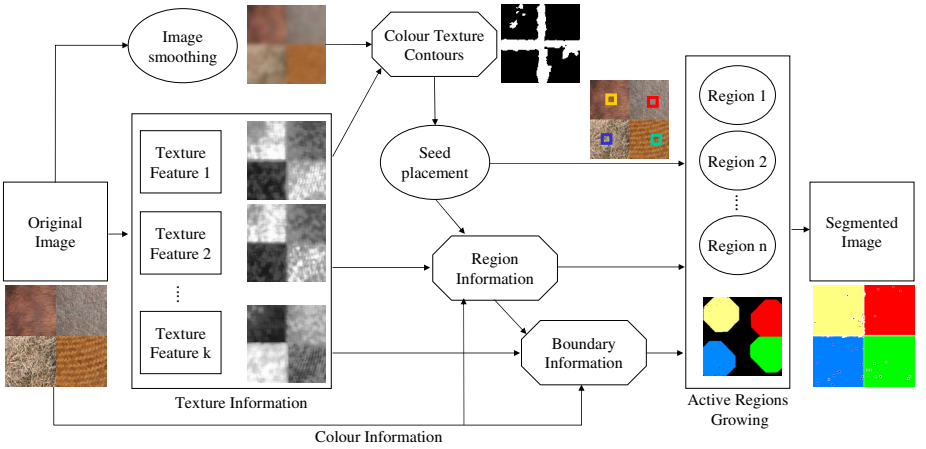



**Fig. 1.** Scheme of the proposed colour texture segmentation strategy.

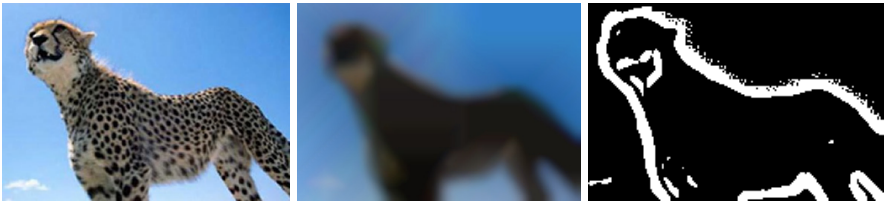
It is only recently that attempts are being made to combine both aspects in a single method. Three alternatives to feature extraction for colour texture analysis appear to be most often used and they consist of: (1) processing each colour band separately by applying grey level texture analysis techniques [10,11], (2) deriving textural information from luminance plane along with pure chrominance features [12,13], and (3) deriving textural information from chromatic bands extracting correlation information across different bands [14,15,16,17].

Our proposal can be classified in the second approach, considering chromatic properties and texture features from the luminance, which facilitates a clear separation between colour and texture features.

## 2 Image Segmentation Strategy

As stated in our review work [5], the different integration strategies try to solve different problems that appear when simple approaches (region or boundary-based) are used separately. Hence, we consider that some of these strategies are perfectly complementary and it could be greatly attractive to fuse different strategies to perform the integration of region and boundary information. The fusion of several approaches will allow to tackle an important number of issues and to exploit at maximum the possibilities offered by each one. Hence, we propose an image segmentation method which combines the *guidance of seed placement*, the *control of decision criterion* and the *boundary refinement* approaches.

Our approach uses the perceptual edges of the image to adequately place a set of seeds in order to initialise the active regions. The knowledge extracted on these regions allows to define the region information and to extract accurate boundary information. Then, as these regions grow, they compete for the pixels of the image by using a decision criterion which ensures the homogeneity inside



(a) (b) (c)

**Fig. 2.** Perception of colour textures as homogeneous colour regions when are seen from a long distance. From original textured image (a), the image is progressively blurred until regions appear as homogeneous (b). Next, colour edges can be extracted (c).

the region and the presence of edges at its boundary. A scheme of the proposed strategy is shown in Figure 1. The inclusion of colour texture information into our initial segmentation proposal involves two major issues: 1) the extraction of perceptual edges, 2) the modelling of colour and texture of regions.

### 2.1 Initialisation: Perceptual Edges

To obtain a sample of each region large enough to statistically model its behaviour, initial seeds have to be placed completely inside the regions. Boundary information allows us to extract these positions in the "core" of regions by looking for places far away from contours.

Boundaries between colour texture regions, which are combination of colour edges and texture edges can be considered as perceptual edges, because a human has the ability to detect both ones. The problem of texture edge detection is considered as a classical edge detection scheme in the multidimensional set of $k$ texture features which are used to represent the region characteristics [18]. Meanwhile, the extraction of colour boundaries implies a major difficulty since the use of an edge detector over a colour image produces the apparition of microedges inside a textured region. Our approach, is based on the perception of textures as homogeneous colour regions when they are seen from a long distance [16]. A smoothing process is progressively performed starting from the original image until textures looks homogeneous, as we would look the texture from far away. Then, the application of an edge detector allows to obtain the colour edges. Figure 2 shows the effect of smoothing a textured image; regions which were originally textured are appreciated as homogeneous colour regions.

The union of texture and colour edges provides the perceptual edges of the image. Nevertheless, due to the inherently non-local property of texture and the smoothing process performed, the result of this method are inaccurate and thick contours (see Figure 2.c). However, this information is enough to perform the seed placement in the "core" of regions, which allows to model the characteristics of regions.

### 2.2 Colour Texture Region Information

Colour in a textured region is by definition not homogenous and presents a very variable behaviour through different image regions. Hence, methods which implicitly assume the same shape for all the clusters in the space, are not able to handle the complexity of the real feature space [19]. Therefore, we focus our attention on density estimation from a non-parametric approach since these methods do not have embedded assumptions, and specifically we adopt the kernel estimation technique. Considering colour pixels inside seeds as a set of data points assumed to be a sample of region colour, density estimation techniques allow the construction of an estimate of the probability density function which describes the behaviour of colour in a region. Given $n$ data points $x_i$, $i = 1, \ldots, n$ in the d-dimensional space $R^d$, the multivariate kernel density estimator with kernel $K_H(x)$ and a bandwidth parameter $h$, becomes the expression

$$\hat{f}(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K_H(\frac{x - x_i}{h}) \tag{1}$$

which gives us the probability of a pixel to belong to a region considering colour properties, $P_{R_c}$, on the three-dimensional colour space. Note that in order to use only one bandwidth parameter $h > 0$ the metric of the feature space has to be Euclidean.

On the other hand, texture of each region $R_i$ is modeled by a multivariate Gaussian distribution considering the set of $k$ texture features extracted from the luminance image. Thus, the mean vector $\overrightarrow{\mu_i}$ and the covariance matrix $\Sigma_i$, which are initialised from the seeds, describe the texture homogeneity region behaviour. Therefore, the probability of a pixel of belonging to a region taking textural properties into account, $P_{R_t}$, is given by the probability density function of a multivariate Gaussian distribution.

Considering both properties together, colour and texture, the probability of a pixel $j$ of belonging to a region $R_i$ will be obtained considering the similarity of the colour pixel with the colour of the region, and the similarity of the texture around the pixel with the texture of the region. The combination of both terms gives the equation

$$P_R(j|R_i) = \beta P_{R_c}(j|R_i) + (1 - \beta) P_{R_t}(j|R_i) \tag{2}$$

where $\beta$ weights the relative importance of colour and texture terms to evaluate the region information.

### 2.3 Colour Texture Boundary Information

It is well know that the extraction of boundary information for textured images is a very tougher task. On the other hand, human performance in localising texture edges is excellent, if (and only if) there is a larger patch of texture on each side available. Hence, as Will et al. [20] noted, texture model of the adjacent textures are required to enable precise localisation. The previous initialisation

step of the regions model allows to dispose of this required knowledge and to extract accurate boundary information.

We shall consider that a pixel $j$ constitutes a boundary between two adjacent regions, $A$ and $B$, when the properties at both sides of the pixel are different and fit with the models of both regions. Textural and colour features are computed at both sides (referred as $m$ and its opposite as $n$). Therefore, $P_R(m|A)$ is the probability that features obtained in the side $m$ belong to region $A$, while $P_R(n|B)$ is the probability that the side $n$ corresponds to region $B$. Hence, the probability that the considered pixel is boundary between $A$ and $B$ is equal to $P_R(m|A) \times P_R(n|B)$, which is maximum when $j$ is exactly the edge between textures $A$ and $B$ because textures at both sides fit better with both models.

Four possible neighbourhood partitions (vertical, horizontal and two diagonals) are considered, similarly to the method of Paragios and Deriche [21]. Therefore, the corresponding probability of a pixel $j$ to be boundary, $P_B(j)$, is the maximum probability obtained on the four possible partitions.

### 2.4 Active Region Growing

Recently the concept of active regions as a way to combine both region and boundary information has been introduced. Examples of this approach, called hybrid active regions, are the works of Paragios and Deriche [21], and Chakraborty et al. [22]. This model is a considerable extension on the active contour model since it incorporates region-based information with the aim of finding a partition where the interior and the exterior of the region preserve the desired image properties.

The goal of image segmentation is to partition the image into subregions with homogeneous properties in its interior and a high discontinuity with neighbouring regions in its boundary. With the aim of integrating both conditions, the global energy is defined with two basic terms. Boundary term measures the probability that boundary pixels are really edge pixels. Meanwhile, the region term measures the homogeneity in the interior of the regions by the probability that these pixels belong to each corresponding region. Some complementary definitions are required: let $\rho(R) = \{R_i : i\epsilon[0, N]\}$ be a partition of the image into $\{N + 1\}$ non-overlapping regions, where $R_0$ is the region corresponding to the background region. Let $\partial\rho(R) = \{\partial R_i : i\epsilon[1, N]\}$ be the region boundaries of the partition $\rho(R)$. The energy function is then defined as

$$E(\rho(R)) = (1 - \alpha) \sum_{i=1}^{N} -\log P_B(j : j\epsilon\partial R_i) + \alpha \sum_{i=0}^{N} -\log P_R(j : j\epsilon R_i|R_i)) \quad (3)$$

where $\alpha$ is a model parameter weighting both terms: boundary and region. This function is then optimised by a region competition algorithm [23] which takes the neighbouring pixels to the current region boundaries $\partial\rho(R)$ into account to determine the next movement. Specifically, a region aggregates a neighbouring

pixel when this new classification decreases the energy of the segmentation. Intuitively, all regions begin to move and grow, competing for the pixels of the image until an energy minimum is reached. When the optimisation process finishes, if there is a background region $R_0$ which remains without being segmented, a new seed is placed in the core of the background and the energy minimisation starts again. This step allows a correct segmentation when a region was missed in the previous stage of initialisation. Furthermore, a final step merges adjacent regions if this causes the energy decrease.

## 3 Experimental Results

Before giving details of the formal method we used to evaluate our proposal, we would like to emphasize a feature that we believe is an important contribution of our proposal: the use of a combination of colour and texture properties. To illustrate this, Figure 3 shows a simple experiment which consists of the segmentation of a mosaic composed by four regions, each of which has a common property, colour or texture, with their adjacent regions. Specifically, each region has the same colour as its horizontal neighbouring region, while the vertical neighbouring region has the same texture. As is stated in the first two examples of Figure 3, the method allows the colour texture properties to be modelled and the four regions to be correctly segmented. On the other hand, we included in Figure 3 a third experiment which consists on the segmentation by using only colour information. Third image shows the smoothed version of the first mosaic image, and clearly illustrates that the original image contains only two colours. Therefore, the segmentation of the image using colour information, although different techniques can be used (considering a Gaussian distribution on the smoothed image, modelling using a mixture of Gaussians on the original image, or other techniques such as the kernel density estimation), will only allow us to identify two colour regions and it is not possible to distinguish regions with the same colour but different texture. As is stated, in some cases colour alone does not provide enough information to perform colour texture analysis, and in order to correctly describe the colour texture of a region, we need to consider not just the colour of pixels, but the relationships between them.

The described segmentation method can be performed over any set of textural features. The result of comparing the relative merits of the different types of features have been nonconclusive and an appropriated set of features has not emerged in all cases. For the experimental trials shown in this article we used the co-occurrence matrices proposed by Haralick et al. [24]. Two of the most typical features, contrast and homogeneity, are computed for distance 1 and for 0°,45°,90° and 135° orientations to constitute a 8-dimensional feature vector. Moreover, the $(L^*, u^*, v^*)$ colour space has been chosen to model the colour.

In order to evaluate the proposed colour texture segmentation technique, we created 9 mosaic images by assembling 4 subimages of size $128 \times 128$ of textures from the VisTex natural scene collection by MIT (*http://www-white.media.mit.edu/vismod/imagery/VisionTexture/vistex.html*), which we have called from $M1$

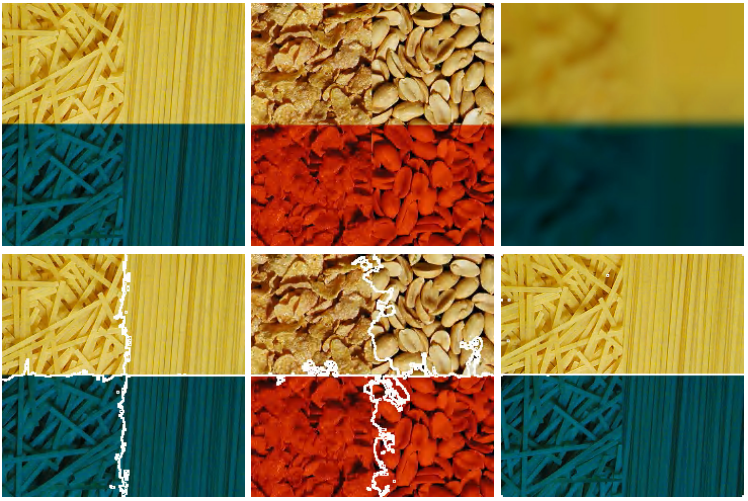


**Fig. 3.** Segmentation of 4 regions composed from two textures and two colours. First row shows the mosaic images. Second row shows the borders of segmented regions.

to $M9$. Furthermore, we added 3 mosaics $M10$, $M11$ and $M12$, provided by Dubuisson-Jolly and Gupta which were used to evaluate their proposal on colour texture segmentation described in [8]. A subset of colour texture mosaic images with obtained segmentation results is shown in Figure 4.

The evaluation of image segmentation is performed by comparing each result with its ground truth and recording the error. Specifically, we use both region-based and boundary-based performance evaluation schemes [25] to measure the quality of a segmentation. Region-based scheme evaluates the segmentation by measuring the percentage of not-correctly segmented pixels considering the segmentation as a multi-class classification problem. Meanwhile, boundary-based scheme evaluates the quality of the extracted region boundaries by measuring the distance from ground truth to the estimated boundary.

Images were processed by our segmentation algorithm using various set of parameter values for the weight of colour (parameter $\beta$) and texture information, as well as the relative relevance of region (parameter $\alpha$) and boundary information in the segmentation process, and best results have been obtained with $\beta = 0.6$ and $\alpha = 0.75$. Note that a predominant role is given to colour and region information. Table 1 shows the quantitative evaluation of results obtained using this parameters setting over the set of mosaic images. Summarising, a mean error of 2.218% has been obtained in the region-based evaluation for the whole set of test images. While the mean error at the boundary has been of 0.841 pixels. Furthermore, our proposal obtained errors of 0.095%, 3.550% and 1.955% in the segmentation of $M10$, $M11$ and $M12$, respectively (see segmentation results of these mosaic images in second row of Figure 4), which can be compared to the

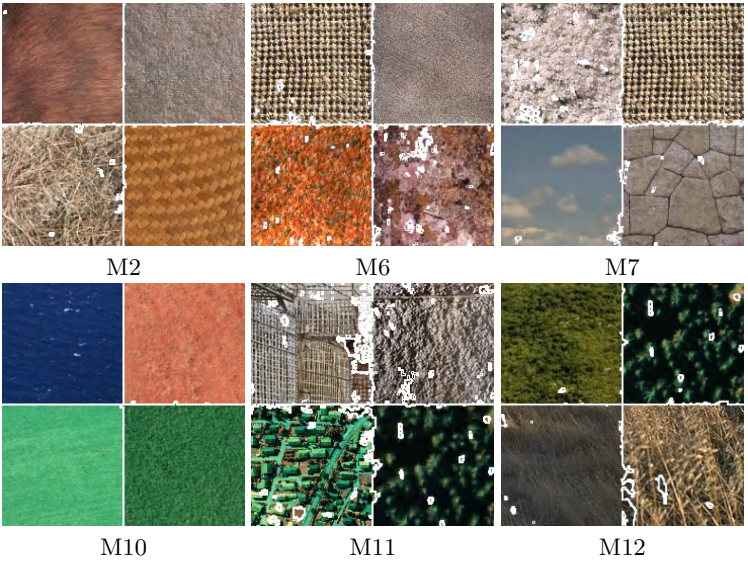



**Fig. 4.** Subset of mosaic colour texture images. Borders of segmented regions are drawn over original images.

segmentation results shown in the work Dubuisson-Jolly and Gupta [8]. Their proposal is a supervised segmentation algorithm based on the fusion of colour and texture segmentations obtained independently. Both segmentations are fused based on the confidence of the classifier in reaching a particular decision. In other words, the final classification of each pixel is based on the decision (from colour or texture) which has obtained a higher confidence. Our results have to be considered as very positive since they significantly improve colour texture segmentation results presented in [8].

Furthermore, the performance of our proposal for colour texture segmentation has been finally tested over a set of real images. Natural scenes predominate among these images, since nature is the most complex and rich source of colour and textures. Some colour texture segmentation results are shown in Figure 5. Meaningful regions in images are successfully detected and the usefulness of our proposal for colour texture segmentation is demonstrated. Furthermore, we want to emphasize some aspects related to the obtained results. See the last example of Figure 5 which shows the segmentation of a monkey among some leaves. The monkey is correctly segmented and, moreover, although the animal is absolutely black several parts of its skin are identified due to their different textural properties. Similar situations occur with other images in which animals are present. In the image with a leopard, region at neck which is not composed by typical spots of the animal, is detected and the same occurs with the lizard image in which the body of the animal, neck and belly are segmented as different regions. It is true that in these cases many of human would group all these regions to com-

**Table 1.** Region-based and boundary-based evaluation for the best results of colour texture segmentation over mosaic images ($\beta = 0.6$ and $\alpha = 0.75$).

| | Region-based (% error) | Boundary-based (pixels distance) |
|---|---|---|
| M1 | 2.207 | 0.352 |
| M2 | 0.280 | 0.145 |
| M3 | 0.731 | 0.237 |
| M4 | 2.375 | 0.588 |
| M5 | 1.663 | 0.786 |
| M6 | 2.352 | 0.341 |
| M7 | 1.451 | 0.596 |
| M8 | 6.344 | 1.774 |
| M9 | 3.609 | 3.430 |
| M10 | 0.095 | 0.028 |
| M11 | 3.550 | 0.962 |
| M12 | 1.955 | 0.852 |
| Mean | 2.218 | 0.841 |
| Std | 1.711 | 0.940 |

pose a single region related to the whole animal body. Nevertheless, this process of assembling is more related to the knowledge that we have about animals that to the basic process of segmentation. Hence, we believe that the segmentation performed by our proposal is correct as it distinguishes regions with different colour texture. The task of region grouping, if necessary, should be carried out by a posterior process which uses higher-level knowledge.

The correctness of boundaries obtained in these segmentations is also shown by the sketch of detected borders over original images. As has been pointed out, texture segmentation is specially difficult at boundaries and great errors are often produced at them. Hence, we want to note the accuracy of segmentations considering not only the correct detection of regions, but also the precise localisation of boundaries between adjacent textures.

## 4 Conclusions

A colour texture image segmentation strategy which integrates region and boundary information has been described. The algorithm uses the contours of the image in order to initialise, in unsupervised way, a set of active regions. Therefore, colour texture of regions is modelled by the conjunction of non-parametric techniques of kernel density estimation and classical texture features. Afterwards, region compete for the pixels optimising an energy function which takes both region and boundary information into account.

The method has been quantitatively evaluated on a set of mosaic images. Furthermore, results over real images riches in colour and texture are shown and

**Fig. 5.** Colour texture segmentation results on real images ($\beta = 0.6$ and $\alpha = 0.75$). Borders of segmented regions are drawn over original images.

analised. The results demonstrate the effectiveness of the proposed algorithm in estimating regions and their boundaries with high accuracy.

**Acknowledgments.** This research was sponsored by the Spanish commission MCyT (Ministerio de Ciencia y Tecnología), and FEDER (Fondo Europeo de Desarrollo Regional) grant TIC2002-04160-C02-01.

Furthermore, we would like to thank M.-P. Dubuisson-Jolly and A. Gupta at Imaging and Visualization Department, Siemens Corporate Research, for providing the texture mosaic images.

## References

1. Pavlidis, T., Liow, Y.: Integrating region growing and edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990) 225–233
2. Haralick, R., Shapiro, L.: Image segmentation techniques. Computer Vision, Graphics and Image Processing **29** (1985) 100–132
3. Pal, N., Pal, S.: A review on image segmentation techniques. Pattern Recognition **26** (1993) 1277–1294
4. Drimbarean, A., Whelan, P.: Experiments in colour texture analysis. Pattern Recognition Letters **22** (2001) 1161–1167
5. Muñoz, X., Freixenet, J., Cufí, X., Martí, J.: Strategies for image segmentation combining region and boundary information. Pattern Recognition Letters **24** (2003) 375–392
6. Muñoz, X., Martí, J., Cufí, X., Freixenet, J.: Unsupervised active regions for multiresolution image segmentation. In: IAPR International Conference on Pattern Recognition, Quebec, Canada (2002)
7. Van de Wouwer, G., Scheunders, P., Livens, S., Van Dyck, D.: Wavelet correlation signatures for color texture characterization. Pattern Recognition **32** (1999) 443–451

8. Dubuisson-Jolly, M.P., Gupta, A.: Color and texture fusion: Application to aerial image segmentation and gis updating. Image and Vision Computing **18** (2000) 823–832
9. Manduchi, R.: Bayesian fusion of color and texture segmentations. In: International Conference on Computer Vision. Volume 2., Corfu, Greece (1999) 956–962
10. Caelli, T., Reye, D.: On the classification of image regions by color, texture and shape. Pattern Recognition **26** (1993) 461–470
11. Thai, B., Healey, G.: Modelling and classifying symmetries using a multiscale opponent colour representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1224–1235
12. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Color and texture-based image segmentation using em and its application to content-based image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 1026–1038
13. Rui, Y., She, A., Huang, T.: Automated region segmentation using attraction-based grouping in spatial-color-texture space. In: IEEE International Conference on Image Processing. Volume 1., Lausanne, Switzerland (1996) 53–56
14. Panjwani, D., Healey, G.: Markov random field models for unsupervised segmentation of textured color images. IEEE Transactions on Pattern Analysis and Machine Intelligence **17** (1995) 939–954
15. Paschos, G.: Fast color texture recognition using chromacity moments. Pattern Recognition Letters **21** (2000) 837–841
16. Mirmehdi, M., Petrou, M.: Segmentation of color textures. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 142–159
17. Tu, Z., Zhu, S.: Image segmentation by data-driven markov chain monte carlo. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 657–673
18. Khotanzad, A., Chen, J.: Unsupervised segmentation of texture images by edge detection in multidimensional features. IEEE Transactions on Pattern Analysis and Machine Intelligence **11** (1989) 414–421
19. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 1–18
20. Will, S., Hermes, L., Buhmann, J., Puzicha, J.: On learning texture edge detectors. In: IEEE International Conference on Image Processing. Volume III., Vancouver, Canada (2000) 887–880
21. Paragios, N., Deriche, R.: Geodesic active regions and level set methods for supervised texture segmentation. International Journal of Computer Vision **46** (2002) 223–247
22. Chakraborty, A., Staib, L., Duncan, J.: Deformable boundary finding influenced by region homogeneity. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Volume 94., Seattle, Washington (1994) 624–627
23. Zhu, S., Yuille, A.: Region competition: Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **18** (1996) 884–900
24. Haralick, R., Shanmugan, K., Dinstein, I.: Texture features for image classification. IEEE Transactions on Systems, Man, and Cybernetics **3** (1973) 610–621
25. Huang, Q., Dom, B.: Quantitative methods of evaluating image segmentation. In: IEEE International Conference on Image Processing. Volume III., Washington DC (1995) 53–56

# Spectral Solution of Large-Scale Extrinsic Camera Calibration as a Graph Embedding Problem

Matthew Brand[1], Matthew Antone[2], and Seth Teller[3]

[1] Mitsubishi Electric Research Labs, 201 Broadway, Cambridge MA 02139 USA,
http://www.merl.com/people/brand/
[2] AlphaTech, 6 New England Executive Park, Burlington MA 01803 USA
[3] MIT, 77 Massachusetts Avenue, Cambridge MA 02139 USA

**Abstract.** Extrinsic calibration of large-scale ad hoc networks of cameras is posed as the following problem: Calculate the locations of $N$ mobile, rotationally aligned cameras distributed over an urban region, subsets of which view some common environmental features. We show that this leads to a novel class of graph embedding problems that admit closed-form solutions in linear time via partial spectral decomposition of a quadratic form. The minimum squared error (MSE) solution determines locations of cameras and/or features in any number of dimensions. The spectrum also indicates insufficiently constrained problems, which can be decomposed into well-constrained rigid subproblems and analyzed to determine useful new views for missing constraints. We demonstrate the method with large networks of mobile cameras distributed over an urban environment, using directional constraints that have been extracted automatically from commonly viewed features. Spectral solutions yield layouts that are consistent in some cases to a fraction of a millimeter, substantially improving the state of the art. Global layout of large camera networks can be computed in a fraction of a second.

## 1 Introduction

Consider a set of images taken from a large number of viewpoints distributed over a broad area such as a city. The source might be a network of security cameras, one or more tourists with cameras, or the collected uploads of camera-enabled mobile phones in a neighborhood. Knowledge of camera positions will be useful for 3D-reconstruction of the environment, tracing the tourist's path, and offering location-aware services to the phone users. We seek to infer these viewpoints from the image set. This paper considers

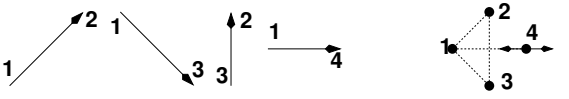



**Fig. 1.** A toy example of the embedding problem. The set of directional constraints at left must be assembled into a maximally consistent graph. The solution (at right) may have degrees of freedom. E.g., node 4 is only partially constrained. In 2D the problem is trivial. In higher dimensions the constraint set may be simultaneously inconsistent, overconstrained, and underconstrained. Our method characterizes and calculates the space of all solutions.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 262–273, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the sparse case, where most viewpoint pairs have nothing in common, but small subsets of cameras view some common environmental features. E.g., in a city, buildings and other large occluders ensure that most images contain local features only.

Antone and Teller [1] demonstrated that it is possible to discover a sparse set of feature correspondences in image sets of this nature. The correspondences can be found from random search or from rough indicators of co-location, e.g., image histogram matches, assignments to wireless communication cells, or global positioning system (GPS) data. Given two internally calibrated cameras that view sufficiently many common scene features, it is possible to determine the *relative* geometry of the cameras, up to scale and orientation [2,3]. Antone and Teller further demonstrated that the *global* orientation of a set of partially overlapped views can be determined from an analysis of feature correspondences and vanishing points in the images [1]. Therefore we have directional information about some of the vectors connecting viewpoints to features (or viewpoints to nearby viewpoints), but no information about distances or locations. This paper introduces a fast spectral method for making a global assignment of viewpoints and features to 3D locations from these sparse directional constraints.

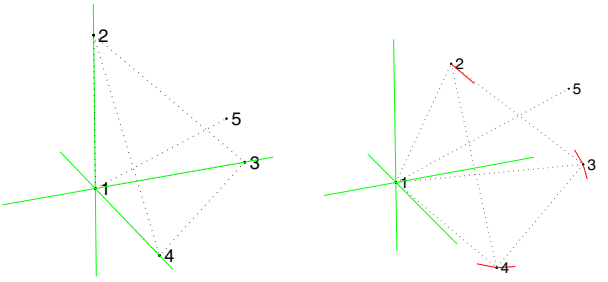
### 1.1 Related Work in Computer Vision

The subject of extrinsic camera calibration has been treated broadly in the computer vision literature [4,2,3]. We review here methods that have decoupled rotational and translational degrees of freedom (DOFs), or have been demonstrated for large or uncertain inputs.

Several researchers have factored the 6-DOF extrinsic calibration problem in order to reduce the number of parameters to be simultaneously estimated. Both interactive [5,6, 7] and automated [8,9] methods exist, and have been demonstrated for relatively small numbers of images. Interactive methods do not scale effectively, and are vulnerable to operator error and numerical instability. Projective techniques [10,11] recover structure and pose only up to an arbitrary projective transformation. Other researchers have described structure-from-motion methods using singular value decomposition [12] or random search [13]. Most of these methods contemplate a much richer set of correspondences than available in our problem. Antone and Teller proposed an iterative algorithm for extrinsic calibration of networks of omni-directional images, using iterative least-squares [1]; each iteration takes time linear in the number of constraints. Section 4 benchmarks our method against theirs.

### 1.2 Related Work in Graph Embeddings

The problem we treat is a particularly well-specified graph embedding problem, and as such bears relation to barycentric embedding proofs of Tutte [14,15] and their modern-day descendent, the locally linear embedding algorithm of Roweis and Saul [16]. These methods constrain node locations to be linear mixtures of their neighbor's locations, with known mixture weights; our method constrains node locations to be linear mixtures of rays emanating from their neighbors, with mixture weights unknown. As it turns out, our solution method has a novel algebraic structure; in contrast to all prior spectral embedding methods, the solution is specified by a single eigenvector, regardless of the dimensionality of the embedding.




| constraint | $1 \to 2$ | $1 \to 3$ | $1 \to 4$ | $2 \to 3$ | $2 \to 4$ | $3 \to 4$ | $1 \to 5$ |
|---|---|---|---|---|---|---|---|
| d$x$ | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| d$y$ | 0 | 1 | 0 | 1 | 0 | -1 | 1 |
| d$z$ | 1 | 0 | 0 | -1 | -1 | 0 | 1 |

**Fig. 2.** A simple embedding problem in $\mathbb{R}^3$. The table of constraints yields a perfect embedding, shown at left with node 1 at the origin, nodes 2-4 on each axis, and node 5 free to slide along a line passing through the middle of the face $\triangle 234$ and the origin. If the constraints are made inconsistent (e.g., by changing the first directional constraint $1 \to 2$ to $[1, 1, 2]^\top$) the least-squares embedding (at right) spreads error evenly over the inconsistent constraints. Node 5 is still free to slide. The small quivers emanating from nodes 2,3,4 show the distances from these nodes to the ray constraints that involve them; the minimal (nontranslational) eigenvalue of $\mathbf{H}_\mathcal{E}$ sums those squared distances.

## 2 Directionally Constrained Embeddings

The directionally constrained embedding problem is formally posed as follows: We are given a set of $N$ nodes $\mathbf{x}_i \in \mathbb{R}^d$and an incomplete specification of node-to-node directions $\mathbf{d}_{ij} \in \mathbb{R}^d$ for some $i, j \in [1, N] \subset \mathbb{N}$, $i \neq j$. The signs and lengths of the true node-to-node displacements are unknown. What is the set of consistent embeddings in $\mathbb{R}^d$? Is the embedding uniquely determined (up to scale and translation)?

We develop a spectral solution based on the thin eigenvalue decomposition (EVD). Let embedding matrix $\mathbf{X} \doteq [\mathbf{x}_1, \cdots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ contain the location $\mathbf{x}_i \in \mathbb{R}^d$ of the $i$th node in column $i$. We seek an embedding where node-to-node displacements $(\mathbf{x}_j - \mathbf{x}_i)$ are maximally consistent with the constraint directions $\mathbf{d}_{ij}$.

### 2.1 Maximum Covariance Spectral Solution

To start simply, first consider maximizing the squared length of the projections of the displacements onto the constraints:

$$\mathcal{C}(\mathbf{X}) \doteq \textstyle\sum_{ij \in \text{constrainted nodes}} \|(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{d}_{ij}\|^2 \,, \tag{1}$$

where $\|\mathbf{X}\|$ denotes Frobenius (Euclidean) norm. Clearly $\mathcal{C}$ is a quadratic form and therefore the problem is convex. To maximize $\mathcal{C}$, consider the vertical concatenation of $N$ embedding vectors $\mathbf{y} \doteq \text{vec}\,\mathbf{X} = [\mathbf{x}_1^\top, \mathbf{x}_2^\top, \cdots \mathbf{x}_N^\top]^\top \in \mathbb{R}^{dN \times 1}$ and the symmetric matrix

$$\mathbf{H}_{\mathcal{C}} \doteq \left[\searrow_i \sum_j \left(\mathbf{d}_{ij}\mathbf{d}_{ij}^\top + \mathbf{d}_{ji}\mathbf{d}_{ji}^\top\right)\right] - \left[\Downarrow_i \left[\Rightarrow_j \left(\mathbf{d}_{ij}\mathbf{d}_{ij}^\top + \mathbf{d}_{ji}\mathbf{d}_{ji}^\top\right)\right]\right] \in \mathbb{R}^{dN\times dN} \quad (2)$$

where $\Downarrow, \Rightarrow, \searrow$ denote vertical, horizontal, and diagonal concatenation, respectively.

**Proposition 1.** *The maximizing eigenvector of* $\mathbf{H}_{\mathcal{C}}$ *determines the embedding* $\mathbf{y}$ *that maximizes* $\mathcal{C}$ *up to sign and scale.*

*Proof.* By construction,

$$\begin{aligned}\mathcal{C} &= \textstyle\sum_{ij} (\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{d}_{ij}\mathbf{d}_{ij}^\top(\mathbf{x}_i - \mathbf{x}_j) \\ &= \textstyle\sum_{ij} \mathbf{x}_i^\top(\mathbf{d}_{ij}\mathbf{d}_{ij}^\top)\mathbf{x}_i + \mathbf{x}_j^\top(\mathbf{d}_{ij}\mathbf{d}_{ij}^\top)\mathbf{x}_j - \mathbf{x}_i^\top(\mathbf{d}_{ij}\mathbf{d}_{ij}^\top)\mathbf{x}_j - \mathbf{x}_j^\top(\mathbf{d}_{ij}\mathbf{d}_{ij}^\top)\mathbf{x}_i \\ &= \mathbf{y}^\top\mathbf{H}_{\mathcal{C}}\mathbf{y}\ .\end{aligned}$$

By the Schmidt-Eckart-Young theorem, the maximum of quadratic form $(\mathbf{y}^\top\mathbf{H}_{\mathcal{C}}\mathbf{y})/(\mathbf{y}^\top\mathbf{y})$ is the largest eigenvalue $\lambda_{\max}(\mathbf{H}_{\mathcal{C}})$, attained at the corresponding eigenvector $\mathbf{y} = \mathbf{v}_{\max}(\mathbf{H}_{\mathcal{C}})$. The optimal embedding is therefore $\mathbf{X} = \mathbf{y}^{(d)} = \mathbf{v}_{\max}^{(d)}$, an order-$d$ vector-transpose that reshapes vector $\mathbf{y} \in \mathbb{R}^{dN}$ into matrix $\mathbf{X} \in \mathbb{R}^{d\times N}$. ■

*Remark 1.* The norm of any directional vector $\|\mathbf{d}_{ij}\|$ determines how strongly it constrains the final solution; if $\|\mathbf{d}_{ij}\| = 0$ the constraint is removed from the problem. Thus we may entertain sparse, weighted constraints.

### 2.2 Minimum Squared-Error (MSE) Spectral Solution

Maximum covariance problems tend to favor solutions in which large displacements are directionally accurate, sometimes at the expense of directionally inaccurate short displacements. A minimum squared-*error* framework is preferable, because it spreads error evenly over the solution, and guarantees an exact solution when allowed by the constraints. To obtain the MSE solution, we minimize the components of the displacements that are orthogonal to the desired directions. The error function is

$$\mathcal{E}(\mathbf{X}) \doteq \textstyle\sum_{ij} \left(\|(\mathbf{x}_i - \mathbf{x}_j)^\top \mathbf{d}_{ij}^\perp\| \cdot \|\mathbf{d}_{ij}\|\right)^2 \quad , \quad (3)$$

where $\mathbf{d}_{ij}^\perp$ is an orthonormal basis of the null-space of $\mathbf{d}_{ij}$. One may visualize each constraint $\mathbf{d}_{ij}$ as a ray emanating from the node $i$ or $j$; $\mathcal{E}$ sums, over all nodes, the squared distance from a node to each ray on which it should lie (scaled by the length of the ray). The constraints are all weighted equally when all $\mathbf{d}_{ij}$ have the same norm. Of course, if the constraints admit an errorless embedding, it will be invariant to any nonzero rescaling of the constraints. To minimize $\mathcal{E}$, let $\mathbf{H}_{\mathcal{E}}$ be constructed as $\mathbf{H}_{\mathcal{C}}$, except that $(\mathbf{d}_{ij}\mathbf{d}_{ij}^\top)$ is replaced everywhere in equation 2 with

$$(\mathbf{d}_{ij}^\top\mathbf{d}_{ij}) \cdot \mathbf{I} - \mathbf{d}_{ij}\mathbf{d}_{ij}^\top\ . \quad (4)$$

**Proposition 2.** *The minimizing eigenvector of* $\mathbf{H}_{\mathcal{E}}$ *in the space orthogonal to* $\mathbf{1}_{[N\times 1]} \otimes \mathbf{I}_{[d\times d]}$ *determines the nondegenerate embedding* $\mathbf{y}$ *that minimizes* $\mathcal{E}$ *up to sign and scale.*

*Proof.* Following the previous proof, $\mathcal{E}(\mathbf{y}) = \mathbf{y}^\top \mathbf{H}_\mathcal{E} \mathbf{y}$ and is minimized by eigenpair $\{\mathcal{E}(\mathbf{y}^{(d)}) = \lambda_{\min}(\mathbf{H}_\mathcal{E}),\ \mathbf{y} = \mathbf{v}_{\min}(\mathbf{H}_\mathcal{E})\}$ because

$$(\mathbf{d}_{ij}^\top \mathbf{d}_{ij}) \cdot \mathbf{I} - \mathbf{d}_{ij}\mathbf{d}_{ij}^\top = (\mathbf{d}_{ij}^\top \mathbf{d}_{ij}) \cdot (\mathbf{I} - \mathbf{d}_{ij}\mathbf{d}_{ij}^\top / (\mathbf{d}_{ij}^\top \mathbf{d}_{ij})) = (\mathbf{d}_{ij}^\perp)\|\mathbf{d}_{ij}\|^2 (\mathbf{d}_{ij}^\perp)^\top,$$

a scaled orthogonal projector that isolates the component of $\mathbf{x}_i - \mathbf{x}_j$ that is orthogonal to $\mathbf{d}_{ij}$ and scales it by $\|\mathbf{d}_{ij}\|^2$. The $\mathbf{1} \otimes \mathbf{I}$ constraint arises because the directional constraints are trivially satisfied by mapping all nodes to a single point in $\mathbb{R}^d$. This implies that the nullspace of $\mathbf{H}_\mathcal{E}$ contains $d$ nuisance eigenvectors that give an orthogonal basis for locating the point anywhere in $\mathbb{R}^d$. Algebraically, that basis is spanned by $\mathbf{1}_{[N\times 1]} \otimes \mathbf{I}_{[d\times d]}$, because $\mathbf{H}_\mathcal{E}(\mathbf{1} \otimes \mathbf{I}) = \mathbf{0}_{[Nd\times d]}$. (This also gives an orthogonal basis for translating the solution in the embedding space). Therefore the nondegenerate embedding must be in the (possibly approximate) nullspace of $\mathbf{H}_\mathcal{E}$ that is orthogonal to $\mathbf{1} \otimes \mathbf{I}$. ■

Appendix A outlines fast and stable methods for computing $\mathbf{y}$ in linear time.

*Remark 2.* Solutions are determined up to sign and scale ($\mathcal{E}(k\mathbf{X}) = k^2\mathcal{E}(-\mathbf{X})$ for $k \in \mathbb{R}^+$). This compares favorably to current methods of spectral graph embeddings, where a $d$-dimensional embedding requires $d$ eigenvectors and is determined only up to arbitrary affine or orthogonal transforms in $\mathbb{R}^d$.

*Remark 3.* (**Uncertain constraints**) If a direction is uncertain, one may replace vector $\mathbf{d}_{ij}$ with a matrix $\mathbf{D}_{ij}$ whose orthogonal columns are each scaled by the certainty that the constraint lies in that direction. The outer product $\boldsymbol{\Sigma}_{ij} \doteq \mathbf{D}_{ij}\mathbf{D}_{ij}^\top$ is effectively the covariance of a Gaussian distribution over possible directions. The associated scaled orthogonal projector is $\|\boldsymbol{\Sigma}_{ij}\|_2 \cdot \mathbf{I} - \boldsymbol{\Sigma}_{ij}$. If the columns of $\mathbf{D}_{ij}$ are unscaled (unit norm), the directional constraint is simply weakened to a subspace constraint, i.e., that $\mathbf{x}_j - \mathbf{x}_i$ lie as close as possible to the subspace spanned by $\mathbf{D}_{ij}$.

## 3 Problem Pathologies

Although the spectral solution is MSE-optimal w.r.t. the constraints, in vision problems the constraints themselves are derived from data and thus may be problematic. Therefore additional tools are needed to detect and resolve ill-posed problems where the constraint data is insufficient or inconsistent.

### 3.1 Underconstrained Problems

A problem is underconstrained when (1) the connectivity graph is disconnected, allowing two partial embeddings that can rigidly transform in each other's coordinate frame, or when (2) all constraints on a node are collinear, allowing it to slide along any one directional constraint. Both cases will manifest themselves as multiple (near-) zero eigenvalues in the spectrum of $\mathbf{H}_\mathcal{E}$; rotating any two eigenvectors in this (approximate) nullspace will animate one such undesired degree of freedom (DOF), giving an *orbit* of solutions. In this way eigenvalue multiplicity diagnoses the dimensionality of the subspace of solutions. However, multiplicity is not 1-to-1 with excess DOFs; an orbit may be redundantly expressed in $d$ eigenvectors, each giving different dynamics for varying the same positional DOF. Some further analysis is needed to characterize the intrinsic DOFs in the problem specification.

When a solution has many DOFs, it is useful to cluster nodes that articulate together. Intuitively, two nodes articulate together if they both have nonzero entries in an eigenvector associated with a zero eigenvalue. Let us call the collection of all such eigenvectors the DOF *matrix*. The optimal clustering is given by the low-dimensional orthogonal *binary* (0/1-valued) basis that best approximates the DOF matrix. Finding such a basis is known to be NP-hard, so we take recourse in continuous relaxations of this problem, e.g., a spectral clustering of the row-vectors of the DOF matrix, or an independent components analysis of its columns. The former seeks groupings whose motions are most decorrelated; the latter seeks full statistical independence, a stronger condition. An example is given below in figure 5. By identifying nonrigidities in the solution, this analysis provides a useful basis for deciding which nodes need additional constraints, e.g., in the form of additional nearby views and associated baseline directions.

### 3.2 Problems Admitting "Negative Lengths"

Because $\mathcal{E}(-\mathbf{X}) = \mathcal{E}(\mathbf{X})$, the projection of a node-to-node displacement onto the desired direction, $(\mathbf{x}_j - \mathbf{x}_i)^\top \mathbf{d}_{ij}$, can be positive *or* negative. In general, perfectly constrained problems will admit solutions only where projections are either all positive or all negative. But problems that are under constrained or that have inconsistent constraints may have MSE solutions with some projections of varied sign. An all-positive solution can be found via quadratic programming (QP). The QP problem has an elegant statement using the eigenvalue decomposition $\mathbf{H}_\mathcal{E} \rightarrow \mathbf{V}\mathrm{diag}(\boldsymbol{\lambda})\mathbf{V}^\top$: We seek a nonzero vector $\mathbf{m} \neq \mathbf{0}$ that mixes the eigenvectors $\mathbf{y} = \mathbf{V}\mathbf{m}$ to incur minimal squared error $\mathcal{E}_{\mathrm{QP}} \doteq \mathbf{m}^\top \mathrm{diag}(\boldsymbol{\lambda})\mathbf{m} = \mathcal{E}(\mathbf{X})$ while keeping all lengths positive. Formally, the QP problem is stated: Minimize $\mathcal{E}_{\mathrm{QP}}$ subject to $\mathbf{m}^\top \mathbf{V}^\top [\Rightarrow_{ij} \mathrm{vec}\, \mathbf{K}_{ij}] \geq \mathbf{1}$, where each $\mathbf{K}_{ij} \in \mathbb{R}^{d \times N}$ is all zeros except for its $i$th and $j$th columns, which are $\mp \mathbf{d}_{ij}$, respectively. Scaling the resulting $\mathbf{m}$ to unit norm is equivalent to performing constrained optimization on a hypersphere. Since eigenvectors paired to large eigenvalues are unlikely to be used, we restrict the problem to consider only the smallest eigenvalue/vector pairs by truncating $\mathbf{V}$ and $\boldsymbol{\lambda}$. This yields a very small quadratic programming problem.

### 3.3 Problems Having Misaligned (Rotated) Nodes

In the *alignment problem* the $i$th node's constraints $\{\mathbf{d}_{ij}\}_j$ are perturbed by a rotation $\mathbf{R}_i^\top$ which we want to identify and remove prior to embedding. Assuming that the directional constraints are consistent with observation data, such perturbations will only be detectable as sources of error in the embedding. Sadly there exist some well-posed embedding problems where rotations induce no errors; see figure 3. Even when rotations do produce error, the alignment problem is almost certainly not convex because the error function is quartic in the rotation parameters (ignoring orthonormality constraints). However, if the perturbations are small and error-producing, it is reasonable to expect that the error function is approximately convex in the neighborhood of the optimal solution, and that

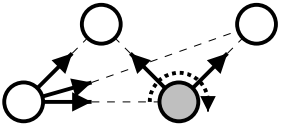


**Fig. 3.** A set of directional constraints (solid arrows) that has a well-constrained embedding for every rotation of the shaded node. The two upper nodes simply slide along the dashed lines.

the MSE embedding is only mildly distorted in this neighborhood. Thus an alternating least-squares procedure that computes embeddings from rotated constraints and rotations from embeddings will iterate to a set of constraints that admits a lower-error embedding. Strictly speaking, this is guaranteed only if one rotation is updated per iteration, but for small perturbations we find that the error declines monotonically when all rotations are updated *en masse*. Solution for the optimal rotation is a Procrustes alignment problem: Collect normalized directional constraints in matrix $\mathbf{A}_i \doteq [\Rightarrow_j \mathbf{d}_{ij}/\|\mathbf{d}_{ij}\|]$ and normalized embedding displacements in matrix $\mathbf{B}_i \doteq [\Rightarrow_j (\mathbf{x}_j - \mathbf{x}_i)/\|\mathbf{x}_j - \mathbf{x}_i\|]$. The optimal aligning rotation $\mathbf{R}_i = \arg\min_{\mathbf{R}\in\mathbb{R}^{d\times d}|\mathbf{R}^\top\mathbf{R}=\mathbf{R}\mathbf{R}^\top=\mathbf{I}} \|\mathbf{R}\mathbf{A}_i - \mathbf{B}_i\|_F$ is $\mathbf{R}_i = \mathbf{V}\mathbf{U}^\top$ from the singular value decomposition $\mathbf{U}\mathrm{diag}(\mathbf{s})\mathbf{V}^\top = \mathbf{A}_i\mathbf{B}_i^\top$. Normalization prevents potential errors due to incorrect displacement lengths.

## 4 Example Calibrations of Camera Networks

The least-squares spectral solution can be implemented in less than fifteen lines of Matlab code. We first illustrate its properties with a simple problem in $\mathbb{R}^3$. Figure 2 shows a perfectly embeddable problem and how the embedding changes when the constraints are made inconsistent. Both problems have a single underconstrained node, so $\lambda_{\min}(\mathbf{H}_\mathcal{E})$ has multiplicity two (ignoring translational DOFs). Rotating the associated eigenvectors causes this node to slide back and forth on its constraint ray while the rest of the solution changes scale to accommodate the algebraic constraint $\|\mathbf{y}\| = 1$.

To assess this method's usefulness on real data with objective performance metrics, we obtained data derived from hundreds of cameras scattered over a university campus [1] and posted to `http://city.lcs.mit.edu`. The datasets consist of 3D directional constraints that were obtained by triangulating pairs of cameras against commonly viewed features of buildings and an analysis of vanishing points in images. The triangulations alone do not give a complete or consistent calibration. The cameras also have rough estimates of ground truth from 2.5-meter-accurate GPS sensors. Other than the noisy GPS data, there is no ground truth for this data. Therefore, we use the self-consistency measures proposed by Antone and Teller to assess the quality of the spectral embeddings as extrinsic camera registrations. Following [1], we report the 3D error of node positions with respect to the directional constraints that apply to them (distance to constraint rays, in millimeters), and the 2D orientation error of node-to-node displacements (angle between displacement and constraint ray, in degrees).

### 4.1 Green Court

The Green Court dataset consists of 32 nodes and 80 directional constraints spanning an area of roughly 80 by 115 meters. The algorithm in [1] recovered global position consistent on average to within 45 millimeters. The maximum position error for any node was 81mm. Reported CPU time of a C implementation was roughly one hour, of which the final layout phase took "a matter of seconds" (personal communication).

Our Matlab implementation computed the optimal (minimum sum-squared error) solution in roughly 1/4 second, reducing consistency errors by roughly four orders of magnitude (see table 1). Figure 4 shows the optimal embedding using the minimizing

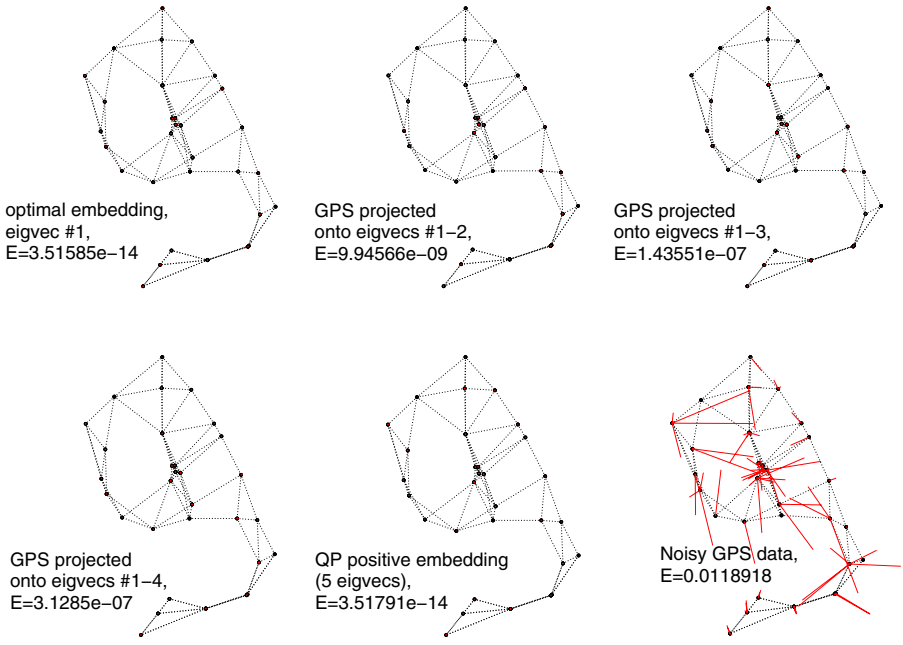


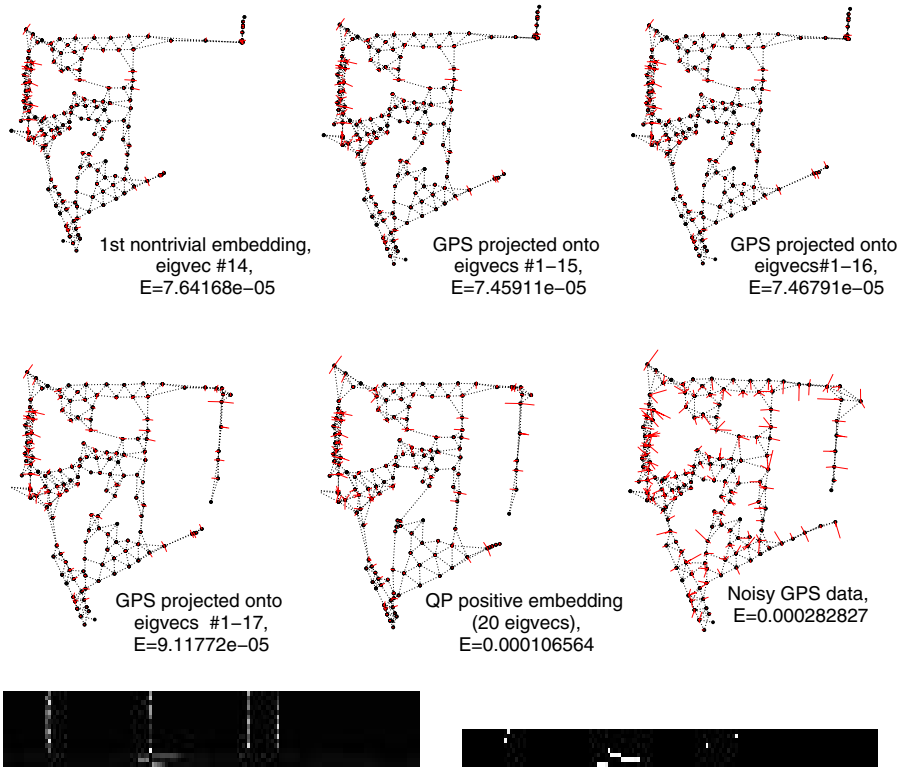

**Fig. 4.** Spectral embedding of the Green Court data, viewed from above. Dotted graph edges indicate problem constraints; each quiver represents the unsatisfied component of a constraint, magnified $10\times$. The rightmost bottom graph shows that GPS errors on the scale of a few meters are being corrected by visually determined constraints. The large holes are building footprints. In this case, the raw spectral solution has strictly positive edge lengths and is thus identical to the QP solution. Note that the spectral embedding has much lower residual ($\mathcal{E}$) than the GPS data. Projecting the GPS data onto low-error embedding subspaces is analogous to data denoising via PCA.

eigenvector and several embeddings obtained by projecting the GPS data onto low-order eigenvectors of $\mathbf{H}$, a form of data denoising analogous to that offered by principal components analysis.

### 4.2 Ames Court

The Ames Court dataset consists of 158 nodes and 355 directional constraints spanning roughly 315 by 380 meters. Antone and Teller report solving a 100-node subset of this problem in which all nodes are properly constrained [1]. They recovered global position consistent on average to within 57mm. The maximum pose inconsistency was 88mm. Reported total CPU time was roughly four hours.

The spectral embedding of the Ames dataset took roughly 3 seconds to compute. Again, the consistency errors are reduced but the results are not directly comparable with those in [1] because the problem is rather different than the version reported in [1]—we have many more nodes and constraints, some of which are inconsistent. For example, the Ames dataset contained one node whose constraints were rotated $43°$ out of




(a) nullspace (DOF) matrix; rows=eigenvectors; cols=nodes

(b) sparse basis for embedding degrees of freedom

**Fig. 5.** Top: Spectral embedding of the Ames Court data, illustrated as in figure 4. Note how the QP solution fixes degrees of freedom in the spectral solution. Subfigure (a) depicts the DOF matrix, transposed such that each row grayscale-codes an eigenvector giving a zero- or low-cost degree of freedom in the embedding. Intensities indicate motility of nodes. Subfigure (b) shows a sparse binary basis for these degrees of freedom obtained by thresholding an independent components analysis of the DOF matrix. Each row indicates a node or group of nodes that can be moved without cost; physically, these are "armatures" of the embedding, often nodes chained by collinear constraints.

alignment with the rest of the data. More notably, our dataset is also underconstrained. This affords an opportunity to illustrate how the spectral analysis identifies degrees of freedom in the solution.

The problem is underconstrained in that several nodes lack more than one linearly independent constraint, and thus can slide freely. It is also overconstrained in that many of the multiply constrained nodes have no error-free embedding; the constraints are slightly inconsistent. Consequently the first 13 eigenvectors give $\mathcal{E} = 0$ embeddings in which the inconsistently constrained nodes are collapsed into point clusters while the rest—mainly

**Table 1.** Consistency errors of the spectral embeddings of Green and Ames datasets. World scale was estimated from GPS baselines.

| Green dataset spectral embedding | mean | max | std |
|---|---|---|---|
| positional error (mm) | $3.541 \times 10^{-3}$ | $1.197 \times 10^{-2}$ | $2.390 \times 10^{-3}$ |
| orientation error (degrees) | $1.823 \times 10^{-5}$ | $1.639 \times 10^{-4}$ | $2.182 \times 10^{-5}$ |

| Ames dataset spectral embedding | mean | max | std |
|---|---|---|---|
| positional error (mm) | $5.696 \times 10^{2}$ | $2.511 \times 10^{3}$ | $3.704 \times 10^{2}$ |
| orientation error (degrees) | $2.280 \times 10^{0}$ | $4.273 \times 10^{1}$ | $3.035 \times 10^{0}$ |

underconstrained nodes—are distributed through space. These "degenerate" solutions turn out to be degrees-of-freedom (DOFs) of the nontrivial solution, eigenvector $\mathbf{v}_{14}$, which is the first eigenvector with nonzero error $\mathcal{E}(\mathbf{v}_{14}^{(3)}) = \lambda_{14} \approx 7.3 \times 10^{-5}$. It specifies an embedding that distributes all nodes through space in a reasonable but imperfect reconstruction of the true scene geometry—reflecting constraints that are not consistent with the true geometry of the scene. Figure 5 shows that the GPS data is well reconstructed by projection to and back-projection from a low-error embedding subspace, comprising the zero-error DOFs (eigenvectors #1-13, which articulate individual nodes), the base solution (eigenvector #14), and a few small-error DOFs (eigenvectors #15-17, which articulate groups of densely connected nodes forming the "arms" of the embedding). Enforcing positive lengths via a small quadratic programming problem automatically finds the correct embedding as a mixture of the 20 lowest-error eigenvectors. This QP embedding has roughly 1/3 the error of the GPS data, and remains stable when more eigenvectors are considered in the QP problem. Rotational alignment further reduces the error $\mathcal{E}$ by roughly an order of magnitude and corrects the $43^\circ$ misaligned node, among others. The rotationally aligned embedding is not diagrammed because it is visually indistinguishable from the QP solution.

### 4.3 Campus

The Campus dataset consists of $566$ nodes and 866 directional constraints spanning roughly a square kilometer. This is far too few to specify an embedding ($\dim(\text{null}(\mathbf{H}_\mathcal{E})) \approx 370$), and indeed this dataset has never been processed as a whole. We used the 400 low-order eigenvectors of $\mathbf{H}_\mathcal{E}$ to denoise this data; figure 6 shows that consistency error declines by an order of magnitude.

## 5 Discussion

The spectral method offers a linear-time optimal solution to graph embedding from directional constraints. These solutions have maximal fidelity to the constraints and provide a basis for identifying flaws in the constraints. Some flaws can be automatically detected and corrected, yielding high-quality embeddings of ill-constrained problems such as the Ames Court dataset even when the constraints have substantial systematic errors. However in general there are classes of ill-posed problems that can be resolved only via additional observations. In the case of the Ames court dataset, where we have a nonrigid embedding that is partitioned into rigid subgraphs and a preferred layout based on positivity constraints, one would propose new views where two subgraphs come

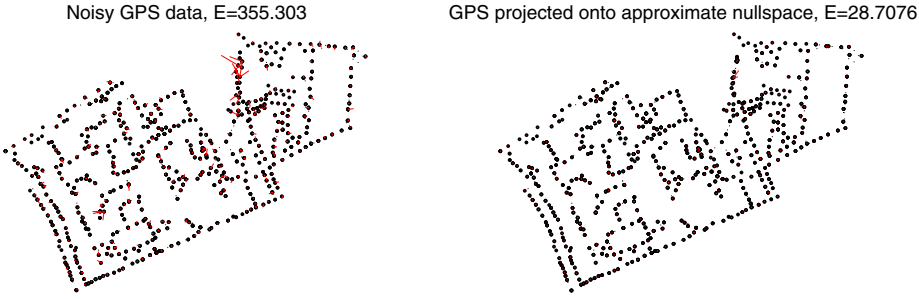



**Fig. 6.** Denoising of MIT GPS dataset via approximate nullspace projection. The group of nodes showing large inconsistencies in the original data (left) are straightened out in the denoised data (right), consistent with the true scene geometry (a street).

close to each other. In that light, one particularly attractive property of spectral schemes is that a solution can be updated in near-linear time when new data arrives, since new nodes and new constraints can be expressed as a series of low-rank modifications to an expanded $\mathbf{H}_{\mathcal{E}}$ matrix. Therefore new viewpoints and environmental features can be added incrementally and efficiently.

## References

1. Antone, M., Teller, S.: Scalable extrinsic calibration of omni-directional image networks. IJCV **49** (2002) 143–174
2. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2000)
3. Faugeras, O., Luong, Q.T., Papadopoulo, T.: The Geometry of Multiple Images. MIT Press, Cambridge, MA (2001)
4. Horn, B.K.P.: Robot Vision. MIT Press, Cambridge, MA (1986)
5. Taylor, C.J., Kriegman, D.J.: Structure and motion from line segments in multiple images. In: Proc. IEEE International Conference on Robotics and Automation. (1992) 1615–1620
6. Becker, S., Bove, V.M.: Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In: Proc. SPIE Image Synthesis. Volume 2410. (1995) 447–461
7. Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In: Proc. SIGGRAPH. (1996) 11–20
8. Shigang, L., Tsuji, S., Imai, M.: Determining of camera rotation from vanishing points of lines on horizontal planes. In: Proc. ICCV. (1990) 499–502
9. Leung, J.C.H., McLean, G.F.: Vanishing point matching. In: Proc. ICIP. Volume 2. (1996) 305–308
10. Mundy, J.L., Zisserman, A., eds.: Geometric Invariance in Computer Vision. MIT Press, Cambridge, MA (1992)
11. Luong, Q.T., Faugeras, O.: Camera calibration, scene motion, and structure recovery from point correspondences and fundamental matrices. IJCV **22** (1997) 261–289
12. Poelman, C.J., Kanade, T.: A paraperspective factorization method for shape and recovery. In: Proc. ECCV. (1994) 97–108
13. Adam, A., Rivlin, E., Shimshoni, I.: ROR: Rejection of outliers by rotations in stereo matching. In: Proc. CVPR. (2000) 2–9

14. Tutte, W.: Convex representations of graphs. Proc. London Mathematical Society **10** (1960) 304–320
15. Tutte, W.: How to draw a graph. Proc. London Mathematical Society **13** (1963) 743–768
16. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290** (2000) 2323–2326

## A Computational Considerations

Present-day numerical eigensolvers may not separate the nullspace of $\mathbf{H}_\mathcal{E}$ into translational and embedding eigenvectors, and in general are prone to numerical error separating the nullspace and near-nullspace eigenvectors. Explicitly suppressing the translational eigenvectors usually improves the numerical stability of the problem. To do so, project $\mathbf{H}_\mathcal{E}$ onto the orthogonal basis $\mathbf{Q} \in \mathbb{R}^{Nd\times(N-1)d}$ of the null-space of the translation basis (i.e., $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$ and $\mathbf{Q}^\top(\mathbf{1}\otimes\mathbf{I}) = \mathbf{0}$), eigen-decompose the reduced problem there, then back-project the eigenvectors:

$$\mathbf{V}'\boldsymbol{\Lambda}\mathbf{V}'^\top \overset{\text{EVD}}{\leftarrow} \mathbf{Q}^\top\mathbf{H}_\mathcal{E}\mathbf{Q} \tag{5}$$

$$\mathbf{V} \leftarrow \mathbf{Q}\mathbf{V}\,. \tag{6}$$

The quadratic form $\mathbf{H}_\mathcal{E}$ is sparse and the null-space basis $\mathbf{Q}$ has a very simple structure, suggesting special computational strategies to defray the cost of computing a very large EVD. In fact, neither matrix need be computed explicity to obtain the desired eigenvector. First we observe that one can use equation (2) to compute $\mathbf{y}^\top(\mathbf{I}-\mathbf{H}_\mathcal{E})$ directly from $\mathbf{y}$ and the directional constraints, thereby yielding a power method for computing the eigenpair $\{\lambda_{\max}(\mathbf{I}-\mathbf{H}_\mathcal{E}),\ \mathbf{v}_{\max}(\mathbf{I}-\mathbf{H}_\mathcal{E})\} = \{1-\lambda_{\min}(\mathbf{H}_\mathcal{E}),\ \mathbf{v}_{\min}(\mathbf{H}_\mathcal{E})\}$ without forming $\mathbf{H}_\mathcal{E}$. Second, note that $\mathbf{Q}$ is a centering matrix: $\mathbf{Q} = \text{null}((\mathbf{1}\otimes\mathbf{I})^\top) = \text{null}(\mathbf{1}^\top)\otimes\mathbf{I}$. The effect of $\mathbf{Q}$ in equations (5–6) is to force the solution to be centered on the origin by ensuring that all rows of $\mathbf{X} = \mathbf{y}^{(d)}$ sum to zero. Equations (5–6) may be dispensed with by modifying the power method to recenter $\mathbf{y}$ on each iteration. This results in an $O(dc)$ time algorithm for $d$ dimensions and $c > N$ constraints. For sparsely constrained problems, complexity is mildly supralinear in the number of nodes ($O(dc) \approx O(dN)$); in a densely constrained problem the complexity will approach $O(dN^2)$ from below.

# Estimating Intrinsic Images from Image Sequences with Biased Illumination

Yasuyuki Matsushita[1], Stephen Lin[1], Sing Bing Kang[2], and Heung-Yeung Shum[1]

[1] Microsoft Research Asia
3F, Beijing Sigma Center, No.49, Zhichun Road Haidian District,
Beijing 100080, China
[2] Microsoft Research
One Microsoft Way, Redmond, Washington 98052-6399, U.S.A.
{yasumat, stevelin, sbkang, hshum}@microsoft.com

**Abstract.** We present a method for estimating intrinsic images from a fixed-viewpoint image sequence captured under changing illumination directions. Previous work on this problem reduces the influence of shadows on reflectance images, but does not address shading effects which can significantly degrade reflectance image estimation under the typically biased sampling of illumination directions. In this paper, we describe how biased illumination sampling leads to biased estimates of reflectance image derivatives. To avoid the effects of illumination bias, we propose a solution that explicitly models spatial and temporal constraints over the image sequence. With this constraint network, our technique minimizes a regularization function that takes advantage of the biased image derivatives to yield reflectance images less influenced by shading.

## 1 Introduction

Variation in appearance caused by illumination changes has been a challenging problem for many computer vision algorithms. For example, face recognition is complicated by the wide range of shadow and shading configurations a single face can exhibit, and image segmentation processes can be misled by the presence of shadows and shading as well. Since image intensity arises from a product of reflectance and illumination, one approach for dealing with variable lighting is to decompose an image into a reflectance component and an illumination component [7], also known as intrinsic images [1]. The reflectance image, free of illumination effects, can then be processed without consideration of shadows and shading.

Decomposition of an image into intrinsic images, however, is an underconstrained problem, so previous approaches in this area introduced additional constraints to make the problem tractable. In [6], it is assumed that the illumination component is spatially smooth while the reflectance component exhibits sharp changes, such that low-pass filtering of the input image yields the illumination image. Similarly, [3] assumes smoothness of illumination and piecewise constant reflectance, so that removing large derivatives in the input image results in the

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 274–286, 2004.
© Springer-Verlag Berlin Heidelberg 2004

illumination image. In addition to illumination smoothness, Kimmel *et al.* [5] include constraints that the reflectance is smooth and the illumination image is close to the input image.

Instead of relying on smoothness constraints, Tappen *et al.* [10] proposed a learning-based approach to separate reflectance edges and illumination edges in a derivative image. Although this method successfully separates reflectance and shading for a given illumination direction used in training, it is difficult to create such a prior to classify edges under arbitrary lighting. Another edge-based method was proposed by Finlayson *et al.* [11] that suppresses color derivatives along the illumination temperature direction to derive a shadow-free image of the scene. In addition to shadow edges, this approach may remove texture edges that also have a color derivative in the illumination temperature direction.

Rather than using only a single input image, Weiss [9] deals with the simpler scenario of having an image sequence captured from a fixed viewpoint with changing illumination conditions. This method employs a ML estimation framework based on a prior that illumination images yield a Laplacian distribution of derivatives between adjacent pixels. Experimental results demonstrate that this technique efficiently and robustly removes cast shadows from reflectance images. Shading on non-planar surfaces, however, can significantly degrade ML estimation of intrinsic images by altering the distribution of derivatives, especially in the typical case of a biased illumination distribution that is not centered around the surface normals of the adjacent pixel pair. More recently, Matsushita *et al.* [12] extended Weiss's method to handle the scenes where the Lambertian assumption does not hold. Using the reflectance image estimated by ML estimation as a scene texture image, their method derives time-varying reflectance images instead of assuming a single reflectance image.

In our proposed method, we also take as input an image sequence and analyze the derivative distributions. Because of the effects of illumination bias on the derivative distributions, we present an alternative method for processing image derivatives, based on explicit modeling of spatial and temporal constraints over the image sequence. With this constraint network, a reflectance image and a set of illumination images are estimated by minimizing a function based on smoothness of illumination and reflectance. Although the derivative distributions are unsuitable for ML estimation, our technique nevertheless takes advantage of derivative distribution information to spatially vary the weight of the smoothness constraints in a manner unlike previous regularization-based methods.

The goal of this work is closely related to that of photometric stereo with unknown light sources and spatially varying albedo. One strong assumption in most uncalibrated photometric stereo approaches [19,20,21] is that there are no cast shadows. However, it is clear that this assumption does not hold in many situations for real world scenes. Yuille *et al.* [22] have proposed a method to handle cast shadows using robust statistics; however, one drawback of the method is that it assumes a single point source in each image. Photometric stereo yields accurate results, but generally it is necessary to assume limiting conditions. While the photometric stereo framework relies on the structural smoothness, our method

relies more on the smoothness of reflectance and illumination images. In the context of photometric stereo, Wolff and Angelopoulou [4] acquired multiple stereo pairs of images of the same scene under different illumination conditions. With two stereo pairs they obtain a stereo pair of photometric ratio images, in which the albedo term is removed in order to extend geometric stereo reconstruction to smooth featureless surfaces.

The remainder of the paper is organized as follows. Sec. 2 details the problem of illumination bias and the resulting effects of shading on derivative distributions. In Sec. 3, we describe the constraints on the energy minimization process and the influence of the derivative distribution. Our algorithm is presented in Sec. 4, followed by experimental results in Sec. 5 and a conclusion in Sec. 6.

## 2 Effect of Illumination Bias

Before describing the effect of illumination bias on derivative distributions, we begin with a brief review of intrinsic images and the ML estimation technique.

Under the Lambertian assumption, as expressed in the following equation, an input image $I$ arises from a product of two intrinsic images: the reflectance image $\rho$ and the illumination image $L$. Since the viewpoint of the image sequence is fixed, $\rho$ does not vary with time $t$. The illumination is comprised by an ambient term $\alpha$ and a direct illumination term $L_D$, which is the product of the illumination intensity $E$, a binary shadow presence function $g$ and the inner product of surface normal $\mathbf{n}$ and illumination direction $\mathbf{l}$:

$$\begin{aligned} I(x,y,t) &= \rho(x,y)L(x,y,t) \\ &= \rho(x,y)\big\{L_D(x,y,t) + \alpha(x,y,t)\big\} \\ &= \rho(x,y)\big\{E(t)g(x,y,t)\big(\mathbf{n}(x,y)\cdot\mathbf{l}(t)\big) + \alpha(x,y,t)\big\} \\ &= \rho(x,y)E(t)\big\{g(x,y,t)\big(\mathbf{n}(x,y)\cdot\mathbf{l}(t)\big) + \alpha'(x,y,t)\big\} \end{aligned} \tag{1}$$

where $\mathbf{n}\cdot\mathbf{l}$ is always non-negative, and $\alpha'$ indicates the ambient term normalized by the illumination intensity $E$. In the ML estimation framework of [9], $n$ derivative filters $f_n$ are first applied to the logarithms of images $I(t)$. For each filter, a filtered reflectance image $\rho_n$ is then computed as the median value in time of $f_n \star \log I$, where $\star$ represents convolution:

$$\log\hat{\rho}_n(x,y) = \mathrm{median}_t\{f_n \star \log I(x,y,t)\}. \tag{2}$$

The filtered illumination images $\log L_n(x,y,t)$ are then computed using the estimated filtered reflectance images $\log\hat{\rho}_n$ according to

$$\log\hat{L}_n(x,y,t) = f_n \star \log I(x,y,t) - \log\hat{\rho}_n(x,y). \tag{3}$$

Finally, a reflectance image $\rho$ and illumination images $L$ are recovered from the filtered reflectance images $\rho_n$ and illumination images $L_n(t)$ through the following deconvolution process,

$$(\log\hat{\rho}, \log\hat{L}) = h \star \Big(\sum_n f_n^r \star (\log\hat{\rho}_n, \log\hat{L}_n)\Big) \tag{4}$$

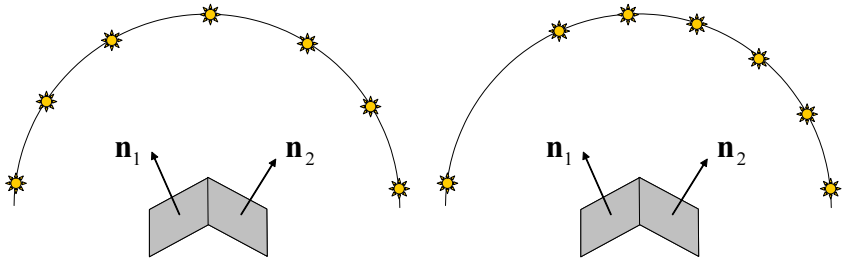



**Fig. 1.** Illumination conditions. (a) Uniform illumination, (b) biased illumination.

where $f_n^r$ is the reversed filter of $f_n$, and $h$ is the filter which satisfies the following equation:

$$h \star \left( \sum_n f_n^r \star f_n \right) = \delta. \tag{5}$$

From (2), it can be shown that for two adjacent pixels with intensities $I_1(t)$ and $I_2(t)$,

$$\hat{\rho}_n = \text{median}_t \frac{I_1(t)}{I_2(t)} = \text{median}_t \frac{\rho_1}{\rho_2} \cdot \frac{E(t)\{g_1 \cdot (\mathbf{n}_1 \cdot \mathbf{l}(t)) + \alpha_1'\}}{E(t)\{g_2 \cdot (\mathbf{n}_2 \cdot \mathbf{l}(t)) + \alpha_2'\}}. \tag{6}$$

We assume that $\alpha'$ is constant over an image, i.e., $\alpha_1'(t) = \alpha_2'(t)$. Cast shadows affect this equation only when $g_1 \neq g_2$. Since this instance seldom occurs, cast shadows do not affect the median derivative values used in ML estimation. It can furthermore be seen that when $\mathbf{n}_1 = \mathbf{n}_2$, shading does not affect ML estimation since $\mathbf{n}_1 \cdot \mathbf{l} = \mathbf{n}_2 \cdot \mathbf{l}$ and consequently $\hat{\rho}_n = \rho_1/\rho_2$.

When a pair of pixels have different surface normals, ML estimation can also deal with shading in cases of *unbiased* illumination samples. For a pair of adjacent pixels with surface normals $\mathbf{n}_1$ and $\mathbf{n}_2$, the set $\Omega_l$ of illumination samples $\mathbf{l}(t)$ are unbiased only under the following condition:

$$\text{median}_{\mathbf{l}(t) \in \Omega_l} \{\mathbf{n}_1 \cdot \mathbf{l}(t) - \mathbf{n}_2 \cdot \mathbf{l}(t)\} = 0. \tag{7}$$

In other words, the illumination is unbiased for a given pair of pixels when the illumination image value $L(x, y)$ of both pixels is the same for the median derivative value. Otherwise, the illumination distribution is biased. Figure 1 shows an illustration of unbiased illumination and biased illumination for a given pair of pixels. With unbiased illumination as given in (7), it can be seen that (6) results in the correct value $\rho_1/\rho_2$.

When a pair of adjacent pixels have different surface normals, illumination bias will cause the ML estimation to be incorrect, because $\mathbf{n}_1 \cdot \mathbf{l} \neq \mathbf{n}_2 \cdot \mathbf{l}$ for the median derivative value. In this case, the illumination ratio in (6) does not equal to one, and consequently $\hat{\rho}_n \neq \rho_1/\rho_2$. This can be expected since different shading is present in the two pixels for every observation.

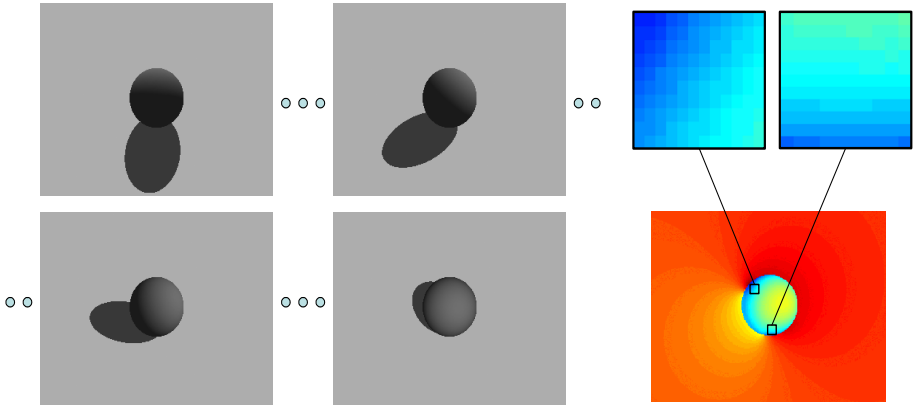


**Fig. 2.** Shading effect remains on reflectance estimate with ML estimation. Left: an input image sequence, Right: the estimated reflectance image with ML estimation.

The case of different surface normals with illumination bias is a significant one, because for a pair of adjacent non-planar pixels, unbiased illumination is rare. So for most pairs of non-planar pixels, ML estimation fails to compute the correct reflectance ratio and the estimated reflectance image contains shading. Figure 2 shows a typical result of ML estimation applied to a synthetic scene with non-planar surfaces. A ball on a plane is lit from various directions as exemplified in the input images on the left side of Figure 2. Although the illumination samples are unbiased for some pairs of pixels, they are biased for most pairs of adjacent pixels. As a result, shading remains in the estimated reflectance image as shown on the right side of the figure.

## 3 Solution Constraints

Since ML estimation is generally affected by shading, we propose an alternative solution method based on the constraints described in this section. Let us denote $i, j$ as labels for illumination directions, $p, q$ for adjacent pixels, and $N, M$ for the number of observed illumination conditions and the number of pixels in an image, respectively.

From a sequence of images, we can derive spatial constraints between adjacent pixels (inter-pixel) and temporal constraints between corresponding pixels under different light directions (inter-frame). Moreover, we employ smoothness constraints to make the problem tractable.

**[Inter-frame constraint]**

Assuming that the scene is composed of Lambertian surfaces, the reflectance value at each point $\rho$ is constant with respect to time. We can thereby derive a temporal constraint as follows:

$$\frac{I_p(t_i)}{I_p(t_j)} = \frac{L_p(t_i)}{L_p(t_j)}, \qquad 0 \leq i, j < N; i \neq j. \tag{8}$$

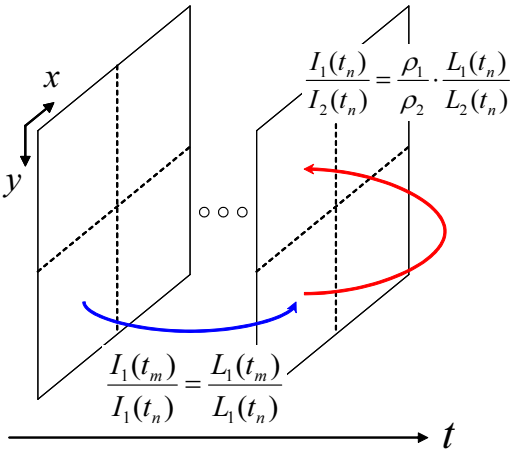



**Fig. 3.** Inter-frame and inter-pixel constraints. A set of constraints composes a constraint network.

This does not determine the absolute values of $L$s; however, it fixes the ratios among $L_p$s.

[**Inter-pixel constraint**]

Letting $\omega_p$ be a set of pixels that are neighbours of $p$,

$$\frac{I_p(t_i)}{I_q(t_i)} = \frac{\rho_p}{\rho_q} \cdot \frac{L_p(t_i)}{L_q(t_i)}, \qquad 0 \leq i < N; q \in \omega_p. \tag{9}$$

This constraint can possibly be applied to non-neighboring pixels, however, we restrict this to be applied only to neighboring pixels because we use the flatness constraint and smooth reflectance constraint in the energy minimization step.

These constraints can be derived from (1), and they compose a 3-D constrained network about $L$ and a 2-D constrained network about $\rho$ as illustrated in Fig. 3. We use them as hard constraints and force $\rho$ and $L$ to always satisfy the following equation:

$$\sum_{p,i,j;i\neq j} \left( \frac{I_p(t_i)}{I_p(t_j)} - \frac{L_p(t_i)}{L_p(t_j)} \right)^2 + \sum_{p,q,i;q\in\omega_p} \left( \frac{I_p(t_i)}{I_q(t_i)} - \frac{\rho_p}{\rho_q} \cdot \frac{L_p(t_i)}{L_q(t_i)} \right)^2 = 0. \tag{10}$$

[**Smoothness constraints**]

In addition to the above constraints, our technique favors smoothness over both $\rho$ and $L$. Smoothness is a generic assumption underlying a wide range of physical phenomena, because it characterizes coherence and homogeneity. Based on the fact that retinal images tend to be smooth according to natural image statistics [16], we assume that both $\rho$ and $L$ are smooth as well. By formulating

these two assumptions into an energy function, we derive intrinsic images by an energy minimization scheme.

The choice of energy function $E(\rho, L)$ is a critical issue. Various kinds of energy functions that measure the smoothness of observed data have been proposed. For example, in regularization-based methods [13,14], the energy minimization function makes the observed data smooth everywhere. This generally yields poor results at object boundaries since discontinuities should be preserved. One discontinuity-preserving function is Geman and Geman's MRF-based function [15].

Although our method assumes smoothness of $L$, this condition clearly does not hold when the surface normals of adjacent pixels differ. In such instances, the smooth $L$ constraint should be weakened. To estimate the amount of difference between neighboring surface normals, we use the information present in the derivative distribution.

If a pair of pixels lie on a flat surface, the values of $I_p(t)/I_q(t)$ are almost always equal to 1 except when only one of the pixels is shadowed, as discussed in [9]. We use this strong statistic and define an error function based on the hypothesis that $I_p$ and $I_q$ share the same surface normal:

$$e_{pq}(t_i) = \left| \arctan\left\{ \mathrm{median}_t\left(\frac{I_p}{I_q}\right)\right\} - \arctan\left\{\frac{I_p}{I_q}\right\}\right| \tag{11}$$

In (11), $\mathrm{median_t(I_p/I_q)}$ corresponds to ML estimation, which gives the ratio of reflectance if $p$ and $q$ are co-planar. We evaluate the angle between the ratio of reflectance and the ratio of observed intensity to determine if the observation supports the flatness hypothesis. To determine the amount of support for the flatness hypothesis, a threshold $\epsilon$ is used:

$$\xi_{pq}(t_i) = \begin{cases} 1 & \big(e_{pq}(t_i) < \epsilon : accept\big) \\ 0 & \big(e_{pq}(t_i) \geq \epsilon : reject\big) \end{cases} \tag{12}$$

Finally, we compute the ratio of the number of acceptances $\xi$ to the number of total observations $N$ to test the surface-flatness hypothesis. When the surface flatness $f$ is high, it is likely that $L$ is smooth. On the other hand, when $f$ is not high, the smoothness assumption for $L$ should be weakened. We define the flatness $f$ as the square of the acceptance ratio:

$$f_{pq} = \left(\frac{\sum_i \xi_{pq}(t_i)}{N}\right)^2. \tag{13}$$

Using the surface flatness evaluated by (13), we define an energy function $E_\Omega$ to minimize:

$$\begin{aligned} E_\Omega &= \sum_p E_p(\Delta\rho_p, \Delta L_p(t)) \\ &= \sum_p \sum_{q\in\omega_p} \left\{(\rho_p - \rho_q)^2 + \lambda f_{pq}(t_i)\big(L_p(t_i) - L_q(t_i)\big)^2\right\} \end{aligned} \tag{14}$$

where $\lambda$ is a coefficient that balances the smoothness of $\rho$ and $L$.

Equation (14) always converges to a unique solution because $E_\Omega$ is convex with respect to $\rho_p$ and $L_p$. This is confirmed by taking $E_p$'s Hessian $H_p$

$$H_p = \begin{bmatrix} \partial^2 E_p/\partial\rho^2 & \partial^2 E_p/\partial\rho\partial L \\ \partial^2 E_p/\partial L\partial\rho & \partial^2 E_p/\partial \mathrm{L}^2 \end{bmatrix} = \begin{bmatrix} \sum_{q\in\omega_p} 1 & 0 \\ 0 & \lambda\sum_{q\in\omega_p} f_{pq} \end{bmatrix}, \tag{15}$$

the leading principal minors of which are $1 > 0$, $\lambda f_{pq} > 0$ where $\lambda > 0, f_{pq} > 0$, so that the function $E_p$ is strictly convex. Since the sum of convex functions is convex, $E_\Omega$ is also convex because $E_\Omega = \sum_p E_p$.

## 4 Algorithm

With the constraints described in the preceding section, our algorithm can proceed as follows.

**[Step 1 : Initialization]** Initialize $\rho$ and $L$.

**[Step 2 : Hard constraints]** Apply the inter-frame and inter-pixel constraints expressed in (10). Since it is difficult to minimize the two terms in (10) simultaneously, we employ an iterative approach for minimization.

1. Inter-frame constraint. Update $L_p(t_i)$.

$$L_p(t_i) \leftarrow \sum_{j\neq i}\left(\frac{I_p(t_i)}{I_p(t_j)}L_p(t_j)\right)/(N-1) \tag{16}$$

2. Inter-pixel constraint. Update $L_p(t_i)$ and $\rho_p$ with ratio error $\beta$. Letting $M_{\omega_p}$ be the number of $p$'s neighboring pixels,

$$\beta_p(t_i) = \left(\sum_{q\in\omega_p}\frac{I_p(t_i)}{I_q(t_i)}\cdot\frac{\rho_q L_q(t_i)}{\rho_p L_p(t_i)}\right)/M_{\omega_p}. \tag{17}$$

   Since the error ratio $\beta_p(t_i)$ can be caused by some unknown combination of $\rho$ and $L$, we distribute the error ratio equally to both $\rho$ and $L$ in (18) and (20), respectively.

$$L_p(t_i) \leftarrow \sqrt{\beta_p(t_i)}L_p(t_i), \tag{18}$$

$$\beta_p = \left(\sum_i \beta_p(t_i)\right)/N, \tag{19}$$

$$\rho_p \leftarrow \sqrt{\beta_p}\rho_p. \tag{20}$$

3. Return to 1. unless Equation (10) is satisfied.

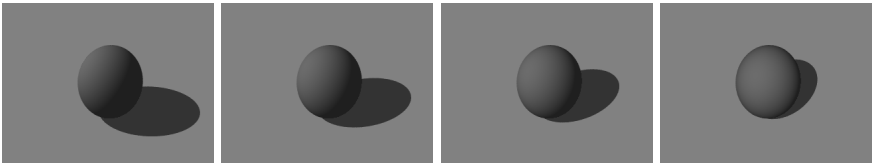


**Fig. 4.** Input image samples from synthetic scene. Illumination samples are chosen to be biased.

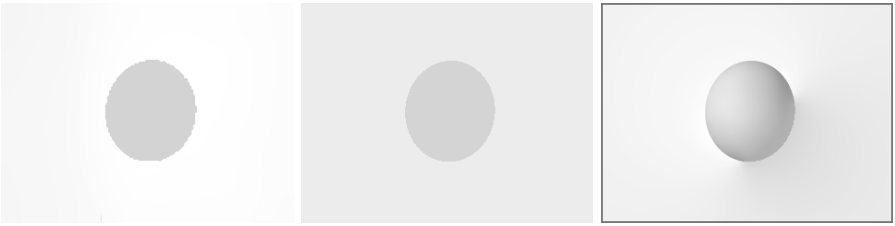


**Fig. 5.** Estimated reflectance images. Left : our method, Center : the ground truth, Right : ML estimation.

**[Step 3 : Energy minimization]**

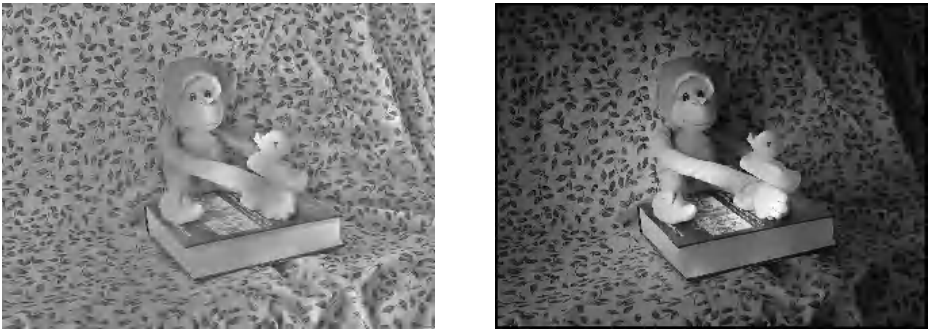
Evaluate the energy function (14), and find $\rho$ and $L$ that lower the total energy. If the total energy can still decrease, update $\rho$ and $L$, then go back to Step 2. Otherwise, we stop the iteration. By fixing $\rho_q$ and $L_q$ in (14), the energy minimization is performed for each $E_p$ using the conjugate gradient method. The conjugate gradient method is an iterative method for solving linear systems of equations which converge faster than the steepest descent method. For further details of the algorithm, readers may refer to a well presented review [18].

## 5 Experimental Results

To evaluate our method, we carried out experiments over one synthetic image sequence and three real world image sequences. In these experiments, we used 11 different lighting conditions, and set $\lambda = 0.4$. We used $\epsilon = 0.02$ which is empirically obtained in Equation (12) for all experiments. The determination of $\epsilon$ has dependency on minimum signal-to-noise ratio. Starting with constant initial values, $\rho$ and $L$ are iteratively updated. There is no restriction about the initial values, however, we used flat images for their initial values because of the smoothness assumption.

### 5.1 Synthetic Scene

For a synthetic scene, we prepared a Lambertian scene with a sphere and a plane as shown in Figure 4. The illumination samples are biased in most cases, since most of them lie on the left-hand side of the images. Figure 5 shows the result of



(a) (b)

**Fig. 6.** Toy scene 1. Estimated reflectance images. (a) Our method, (b) ML estimation.

(a) (b)

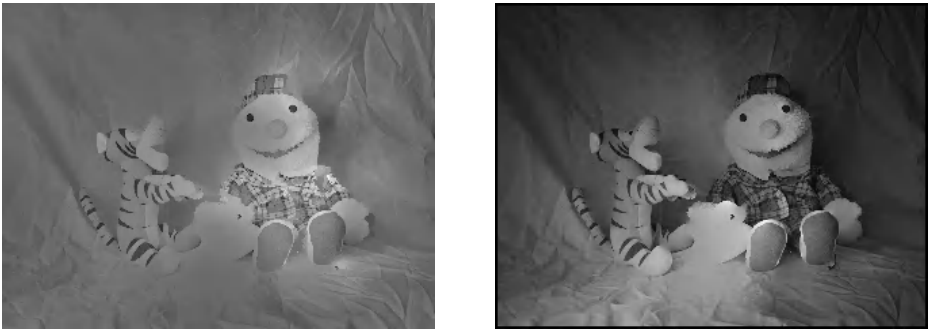
**Fig. 7.** (a) Estimated illumination image, and (b) the corresponding input image.

our method, the ground truth, and the result of ML estimation from left to right. Due to the scaling ambiguity of the reflectance images, we adjusted the scaling of each reflectance image for better comparison and display. As we can clearly see in the figure, our method can successfully derive shading-free a reflectance image which is close to the ground truth.

### 5.2 Real World Scenes

For real world scenes, we captured two image sequences of toy scenes and used Yale Face Database B [17]. Figure 6 and Figure 8 show the result of reflectance estimation from Lambertian scenes. In both figures, the left image shows the result of our method, while the right image shows the result of ML estimation. As we can see clearly in them, our method handles shading more correctly and shading effect is much reduced in our reflectance estimates. Figure 7 and 9 show the estimated illumination images and corresponding input images. In illumination images, reflectance edges such as texture edges are well removed.

On the other hand, Figure 10 shows a negative result, especially on the hair. Since the human hair shows high specularity, and it is hard to model it as Lambertian. This non-Lambertian property affects to our method and turns those area into white. This is because our method is based on Lambertian model, and



(a) (b)

**Fig. 8.** Toy scene 2. Estimated reflectance images. (a) Our method, (b) ML estimation.

(a) (b)

**Fig. 9.** (a) Estimated illumination image, and (b) the corresponding original image.

it implies that our method does not handle specular reflections well. However, as for the non-specular part such as the human face, shading effect is much reduced by our method.

## 6 Conclusion

We have presented a method that robustly estimates intrinsic images by energy minimization. Unlike previous methods, our method is not affected by illumination bias which generally exists. In our framework, we explicitly modeled spatial and temporal constraints over the image sequence to form a constraint network. Using this as a hard constraint, we minimized an energy function defined from the assumptions that reflectance and illumination are smooth. By weighting these smoothness constraints according to a surface flatness measure estimated from derivative distributions, we estimated intrinsic images with improved handling of shading. Evaluation with both synthetic and real world image sequences shows that our method can robustly estimate shading-free reflectance image and illumination images. Some of the next steps of our research will include the acceleration of energy minimization part and extension of our model to correctly handle specularity.

(a) (b)

**Fig. 10.** Reflectance images estimated from Yale Face Database B. (a) Our method, (b) ML estimation. The result shows the limitation of our method, i.e. high specularity affects the result.

## References


1. H.G. Barrow and J.M. Tenenbaum: Recovering intrinsic scene characteristics from images. In *A. Hanson and E. Riseman, editors, Computer Vision Systems. Academic Press*, 3–26, 1978.
2. E.H. Adelson and A.P. Pentland: The Perception of Shading and Reflectance. In *D. Knill and W. Richards (eds.), Perception as Bayesian Inference*, 409–423, 1996.
3. A. Blake: Boundary conditions of lightness computation in mondrian world. In *Computer Vision, Graphics and Image Processing* , 32, 314–327, 1985.
4. L. B. Wolff and E. Angelopoulou: 3-d stereo using photometric ratios. In *Proc. of the Third European Conference on Computer Vision (ECCV)*, pp 247–258, 1994.
5. R. Kimmel, M. Elad, D. Shaked, R. Keshet and I. Sobel: A Variational Framework for Retinex. In *International Journal of Computer Vision*, 52(1), 7–23, 2003.
6. E.H. Land: An alternative technique for the computation of the designor in the Retinex theory of color vision. In *Proc. Nat. Acad. Sci.* , 83, 3078–3080, Dec. 1986.
7. E.H. Land: The Retinex theory of color vision. In *Scientific American* , 237(G), No. 6, 108–128, Dec. 1977.
8. E.H. Land, and J.J. McCann: Lightness and retinex theory. In *Journal of the Optical Society of America*, 61(1), 1–11, 1971.
9. Y. Weiss: Deriving intrinsic images from image sequences. In *Proc. of 9th IEEE Int'l Conf. on Computer Vision*, 68–75, Jul., 2001.
10. M.F. Tappen, W.T. Freeman, and E.H. Adelson: Recovering Intrinsic Images from a Single Image. In *Advances in Neural Information Processing Systems 15 (NIPS)*, MIT Press, 2002.
11. G.D. Finlayson, S.D. Hordley, and M.S. Drew: Removing Shadows from Images. In *Proc. of European Conf. on Computer Vision* Vol.4, 823–836, 2002.
12. Y. Matsushita, K. Nishino, K. Ikeuchi and M. Sakauchi : Illumination Normalization with Time-dependent Intrinsic Images for Video Surveillance. In *Conf. on Computer Vision and Pattern Recognition (CVPR)*, Vol.1, pp. 3–10, 2003.
13. A. Blake: Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):2–12, 1989.
14. B.K.P. Horn and B. Schunk: Determining optical flow. In *Artificial Intelligence*,17:185–203, 1981.

15. S. Geman and D. Geman: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.
16. B.A. Olshausen and D.J. Field: Emergence of simplecell receptive field properties by learning a sparse code for natural images. In *Nature*, 381:607-608, 1996.
17. A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman: From few to many: Generative models for recognition under variable pose and illumination. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, 277-284, 2000.
18. J.R. Shewchuck: An introduction to the conjugate gradient method without agonizing pain. *Tech. Rep. CMU-CS-94-125, Carnegie Mellon University*, 1994.
19. H. Hayakawa: Photometric stereo under a light-source with arbitrary motion. In *Journal of Optical Society of America A.*, 11(11):3079–3089, 1994.
20. R. Basri and D. Jacobs: Photometric stereo with general, unknown lighting. In *Proc. of Computer Vision and Pattern Recognition(CVPR)*, Vol.2, pp. 374–381, 2001.
21. A.S. Georghiades, P.N. Belhumeur and D.J. Kriegman: Illumination-Based Image Synthesis: Creating Novel Images of Human Faces Under Differing Pose and Lighting. In *Proc. Workshop on Multi-View Modeling and Analysis of Visual Scenes*, pp. 47–54, 1999.
22. A.L. Yuille, D. Snow, R. Epstein, P. Belhumeur: Determining Generative Models for Objects Under Varying Illumination: Shape and Albedo from Multiple Images Using SVD and Integrability. In *International Journal on Computer Vision.*, 35(3), pp 203–222. 1999.

# Structure and Motion from Images of Smooth Textureless Objects

Yasutaka Furukawa[1], Amit Sethi[1], Jean Ponce[1], and David Kriegman[2]

[1] Beckman Institute, University of Illinois at Urbana-Champaign
{yfurukaw,asethi}@uiuc.edu, ponce@cs.uiuc.edu
[2] Dept. of Computer Science, University of California at San Diego
kriegman@ucsd.edu

**Abstract.** This paper addresses the problem of estimating the 3D shape of a smooth textureless solid from multiple images acquired under orthographic projection from unknown and unconstrained viewpoints. In this setting, the only reliable image features are the object's silhouettes, and the only true stereo correspondence between pairs of silhouettes are the *frontier points* where two viewing rays intersect in the tangent plane of the surface. An algorithm for identifying geometrically-consistent frontier points candidates while estimating the cameras' projection matrices is presented. This algorithm uses the *signature* representation of the dual of image silhouettes to identify promising correspondences, and it exploits the redundancy of multiple epipolar geometries to retain the consistent ones. The visual hull of the observed solid is finally reconstructed from the recovered viewpoints. The proposed approach has been implemented, and experiments with six real image sequences are presented, including a comparison between ground-truth and recovered camera configurations, and sample visual hulls computed by the algorithm.

## 1 Introduction

Structure and motion estimation algorithms typically assume that correspondences between *viewpoint-independent* image features such as interest points or surface markings have been established via tracking or some other mechanism (e.g., [4,21,23]). Several effective techniques for computing a projective, affine, or Euclidean scene representation from these correspondences while estimating the corresponding projection matrices are now available (see, for example [8,9, 13] for extensive discussions of such methods). For objects with little texture and few surface markings, silhouettes are the most reliable image features. The silhouette of a smooth solid is the projection of a surface curve, the *occluding contour*, where the viewing cone grazes the surface. Establishing correspondences between these *viewpoint-dependent* features is difficult: In fact, there is only a finite number of true stereo correspondences between any two silhouettes, namely the *frontier points* where the two occluding contours and the corresponding viewing rays intersect in the tangent plane of the surface [10].

For image sequences taken by a camera with known motion, it is possible to estimate the second-order structure of a surface along its occluding contour,

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 287–298, 2004.
© Springer-Verlag Berlin Heidelberg 2004

as first shown by Giblin and Weiss in the orthographic projection case [12] (see, for example, [5,7,20] for extensions to perspective projection). Methods for recovering both the surface structure and the camera motion using a trinocular rig have also been proposed [14,25]. The single-camera case is more difficult, and all approaches proposed so far have either been limited to circular motions [11, 18,28], required a reasonable guess to bootstrap an iterative estimation process [2,6], or been limited to synthetic data [26]. Likewise, all published methods for computing visual hulls [16] from image silhouettes, dating back to Baumgart's 1974 thesis [3], have assumed that the camera configurations were known a priori.

This paper presents an integrated approach to the problem of estimating both structure and motion for smooth textureless solids observed by orthographic cameras with unknown and unconstrained viewpoints. An algorithm for identifying geometrically-consistent frontier point candidates while estimating the cameras' projection matrices is presented. This algorithm uses the *signature* representation of the dual of image silhouettes, proposed in [1] in the object recognition context, to identify promising correspondences, and it exploits the redundancy of multiple epipolar geometries [17] to retain the consistent ones. The visual hull [3,16] of the observed solid is finally reconstructed from the recovered viewpoints. We have implemented this algorithm, and tested it on six real image sequences.

## 2 Proposed Approach

As mentioned in the previous section, the only true stereo correspondences between two silhouettes of a smooth solid are a finite number of *frontier points*, where two viewing rays intersect as they graze the surface along the same tangent plane (Figure 1). Equivalently, the frontier points are the intersections of the corresponding occluding contours on the surface.

As will be shown in Section 2.2, it is a relatively simple matter to estimate the projection matrices associated with $m$ views of a smooth surface when a sufficient number of *true* frontier points are available for a sufficient number of image pairs. Conversely, it is easy to find the frontier points associated with a pair of images once the corresponding projection matrices are known since the corresponding tangent lines run parallel to the epipolar lines. This suggests the following algorithm for robustly estimating the projection matrices while identifying correct matches between silhouette pairs. It is similar in spirit to the RANSAC-based approach to weak calibration proposed in [22].

1. For each image pair, select a set of promising frontier points candidates. Each candidate will be referred to as a *match* between the two images in the sequel.
2. Find a minimal set of images and geometrically-consistent matches, and estimate the corresponding pairwise epipolar geometries and the individual projection matrices;
3. Add the remaining images one by one, using matches that are geometrically consistent with the current set of images to estimate the corresponding projection matrices.

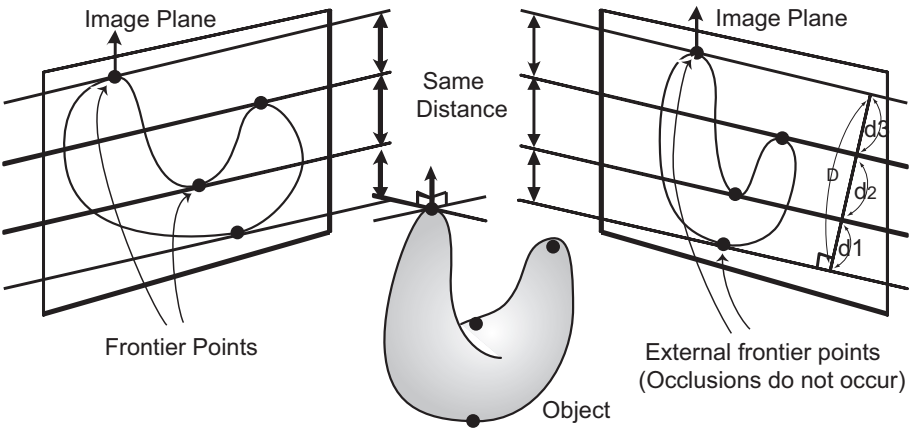



**Fig. 1.** Frontier points. See text for details.

Three main ingredients play a role in the successful implementation of this algorithm—namely, effective techniques for (1) selecting promising matches between pairs of images; (2) estimating the projection matrices from these matches; and (3) rejecting matches that are not consistent with all available geometric information. These ingredients are detailed in the following sections.

### 2.1 Selecting Frontier Point Candidates

A fundamental property of frontier points under orthographic projection is that the tangent lines at these points are parallel to each other, and the distances between successive tangents are the same in the two images. This property was used in [1] as the basis for a 3D object recognition algorithm. Briefly, the *signature* of a planar curve $\Gamma$ is defined by mapping every unit vector $\boldsymbol{n}$ in the plane onto the tuple formed by the successive distances between the tangent lines to $\Gamma$ perpendicular to $\boldsymbol{n}$ (Figure 1), taken in the order in which they are traversed by that vector. Formally, the signature can be thought of as a representation of the set of tangent lines—or *dual*—of $\Gamma$ by a family of curves embedded in subspaces of $\mathbb{R}^d$ of various dimensions, where $d$ is the maximum number of parallel tangents of $\Gamma$ [1]. In the structure-from-motion context, this interpretation is not necessary. Instead, it is sufficient to note that the signatures of two silhouettes intersect at the corresponding frontier points, which affords a simple mechanism for selecting potential pairs of frontier points.

To account for the possibility of self occlusion, we follow the robust matching approach of [1,24] to determine the "distance" between two signature points $\boldsymbol{d} = (d_1, \ldots, d_k)$ and $\boldsymbol{d'} = (d'_1, \ldots, d'_l)$, where $k$ may not equal $l$. Assuming that $d_{ij} = |d_i - d'_j|$ obeys a normal distribution with variance $\sigma$ for matching entries, and a uniform distribution for all others, the discrepancy between individual entries in $\boldsymbol{d}$ and $\boldsymbol{d'}$ is the Lorentzian $L_\sigma = \sigma^2/(d_{ij}^2 + \sigma^2)$, whose value is 1 for a perfect match but is close to zero for large mismatches. To respect the natural ordering of the tangent lines, the final score is found by using dynamic

programming to maximize the sum of the Lorentzians among all paths with non-decreasing function $j(i)$, and dividing the maximum by the number of matched signature points.

This approach provides a guide for selecting promising matches. We also use a number of filters for rejecting incorrect ones: First, the object should lie on the same side of matching tangents in both images. Second, the curvatures at matching frontier points should have the same sign [15]. In practice, we exhaustively search each pair of silhouettes for potential sets of frontier points,[1] and retain the $t$ most promising ones, where $t$ is a fixed constant ($t = 10$ in our implementation).

### 2.2 Estimating Projection Matrices from Frontier Points

We assume an affine (orthographic, weak-perspective, or para-perspective) projection model, and show in this section how to estimate the projection matrices associated with a set of cameras from the corresponding object silhouettes and their pairwise frontier points. Contrary to the typical situation in structure from motion, where many point features are visible in many images, a (relatively) small set of frontier points is associated with each pair of images, and it is only visible there. Therefore, a different approach to motion estimation is called for. We proceed in three steps as described below.

**Affine motion from a pair of images.** Exploiting the affine ambiguity of affine structure from motion allows us to write the projection matrices associated with two images $I$ and $I'$ in the *canonical form* (see [9] for example):

$$\hat{\mathcal{M}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \hat{\mathcal{M}}' = \begin{bmatrix} 0 & 0 & 1 & 0 \\ a & b & c & d \end{bmatrix}. \tag{1}$$

Assuming there are $n$ frontier points with three-dimensional coordinates $(x_j, y_j, z_j)$ and image coordinates $(u_j, v_j)$ and $(u'_j, v'_j)$ $(i = 1, \ldots, n)$, it follows immediately that

$$au_j + bv_j + cu'_j - v'_j + d = 0 \text{ for } j = 1, \ldots, n. \tag{2}$$

This is of course equivalent to the affine epipolar constraint $\alpha u_j + \beta v_j + \alpha' u'_j + \beta' v'_j + \delta = 0$, where the coefficients $a$, $b$, $c$, and $d$ are related to the parameters $\alpha$, $\beta$, $\alpha'$, $\beta'$, and $\delta$ by $a : \alpha = b : \beta = c : \alpha' = -1 : \beta' = d : \delta$. Given the images of $n$ frontier points, the parameters $a$, $b$, $c$, and $d$ can be computed by using linear least squares to solve the over-constrained system of linear equations (2) in these unknowns.

[1] We could of course use some hashing technique—based, say, on the diameter $D$ of the object in the direction of interest—to improve the efficiency of the search for promising matches, but this is far from being the most costly part of our algorithm.

**Affine motion from multiple images.** This section shows how to recover the $m$ projection matrices $\mathcal{M}_i$ $(i = 1, \ldots, m)$ in some global affine coordinate system once the pairwise epipolar geometries are known, or, equivalently, once the projection matrices are known in the canonical coordinate systems attached to each camera pair.

Suppose that the values $(a_{kl}, b_{kl}, c_{kl}, d_{kl})$ associated with two images $I_k$ and $I_l$ have been computed from (2). There must exit some affine transformation $\mathcal{A}$ mapping the canonical form (1) onto $\mathcal{M}_k$ and $\mathcal{M}_l$, i.e.,

$$\begin{bmatrix} \mathcal{M}_k \\ \mathcal{M}_l \end{bmatrix} = \begin{bmatrix} \hat{\mathcal{M}}_k \\ \hat{\mathcal{M}}_l \end{bmatrix} \mathcal{A}. \tag{3}$$

If we write the two projection matrices $\mathcal{M}_k$ and $\mathcal{M}_l$ as

$$\mathcal{M}_k = \begin{bmatrix} \boldsymbol{p}_k^T \\ \boldsymbol{q}_k^T \end{bmatrix} \quad \text{and} \quad \mathcal{M}_l = \begin{bmatrix} \boldsymbol{p}_l^T \\ \boldsymbol{q}_l^T \end{bmatrix},$$

it is a simple matter to eliminate the unknown entries of $\mathcal{A}$ in Eq. (3) and show that

$$\boldsymbol{q}_l = \begin{bmatrix} \boldsymbol{p}_k & \boldsymbol{q}_k & \boldsymbol{p}_l & \mathbf{0} \end{bmatrix} \boldsymbol{e}_{kl},$$

where $\mathbf{0} = (0, 0, 0, 1)^T$, and $\boldsymbol{e}_{kl} = (a_{kl}, b_{kl}, c_{kl}, d_{kl})^T$. In other words, we have four linear constraints on the entries of the matrices $\mathcal{M}_k$ and $\mathcal{M}_l$. By combining the equations associated with all image pairs, we obtain a linear system of $2m(m-1)$ linear equations in the $8m$ entries of the $m$ projection matrices, whose solutions are only defined up to an arbitrary affine transformation. We remove this ambiguity by fixing two projection matrices to their canonical form given by (1). The solution of the remaining $p = 2m(m-1) - 4$ linear equations in $q = 8(m-2)$ unknowns is again computed by using linear least squares. Three images are sufficient to compute a single solution, and four images yield redundant equations that can be used for consistency checks as explained in the next section.

**Euclidean motion.** Let us write the affine projection matrices recovered in the previous section as $\mathcal{M}_i = \begin{bmatrix} \mathcal{A}_i & \boldsymbol{b}_i \end{bmatrix}$ $(i = 1, \ldots, m)$. As shown in [19] for example, once the affine projection matrices are known, there exists an affine transformation, or *Euclidean upgrade*,

$$\mathcal{Q} = \begin{bmatrix} \mathcal{C} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \text{ such that } \mathcal{M}_i \mathcal{Q} = \begin{bmatrix} \mathcal{R}_i & \boldsymbol{b}_i \end{bmatrix},$$

where the $2 \times 3$ matrix $\mathcal{R}_i$ is the top part of a $3 \times 3$ rotation matrix and, this time, $\mathbf{0} = (0, 0, 0)^T$. It follows that $\mathcal{A}_i(\mathcal{C}\mathcal{C}^T)\mathcal{A}_i^T = \mathcal{A}_i \mathcal{S} \mathcal{A}_i^T = \mathrm{Id}_2$, where $\mathcal{S} = \mathcal{C}\mathcal{C}^T$, and $\mathrm{Id}_2$ is the $2 \times 2$ identity matrix. The $m$ instances of this equation provide $3m$ constraints on the 6 independent entries of the symmetric matrix $\mathcal{S}$, allowing its

recovery via linear least squares. Once $\mathcal{S}$ is known, the matrix $\mathcal{C}$ can be recovered using Cholesky factorization for example.[2]

### 2.3 Enforcing Geometric Consistency

As shown in [17] for example, the pairwise epipolar constraints among a set of images are redundant. We propose in this section to exploit this redundancy by enforcing the corresponding geometric constraints during matching.

**Geometric consistency constraints.** The following simple tests can be used to check whether a set of matches and the corresponding projection matrices are geometrically consistent:

1. *Motion estimation residuals.* As shown in Section 2.2, the recovery of the affine projection matrices from a set of frontier points can be formulated as a linear least-squares problem. The size of the corresponding residual gives a first measure of consistency. The same is true of the residual of the linear system associated with the corresponding Euclidean upgrade. We use both measures in our implementation as simple filters for rejecting incorrect matches.
2. *Unmatched external frontier points.* Suppose the projection matrices associated with $m$ images have been estimated, but matches of some image pairs $(I_k, I_l)$ have *not* been used in the estimation process (this is a typical situation because of the epipolar constraints' redundancy). The affine fundamental matrix associated with $I_k$ and $I_l$ is easily computed from the corresponding projection matrices, and it can be used to predict the frontier points' projections in both images. Due to noise, discretization errors, occlusions, etc., some of the predicted points in one image may not have matches in the other one. Still, the two outermost—or *external*—frontier points are normally visible in each image (Figure 1), even in the presence of self occlusion, and they can be used as a second consistency filter. Of course, the distance between these points should be the same in the two images, i.e., the diameters of the two silhouettes in the direction orthogonal to the epipolar lines should be the same. But one can go further and compute the distance separating each external frontier point from the epipolar line associated with its match. This test, that computes four images distances instead of a single diameter difference, has proven much more discriminative in our experiments.
3. *Matched frontier points.* Assuming as before that the projection matrices are known, the 3D positions of all matched frontier points are easily reconstructed via triangulation. Our third consistency check is to project these frontier points into every other image and see if they lie outside the corresponding silhouette. Sum of distances of outlying frontier points to the closest point on each silhouette becomes the measure.

[2] This assumes that $\mathcal{S}$ is positive definite, which may not be the case in the presence of noise. See [21] for another approach based on non-linear least squares and avoiding this assumption.

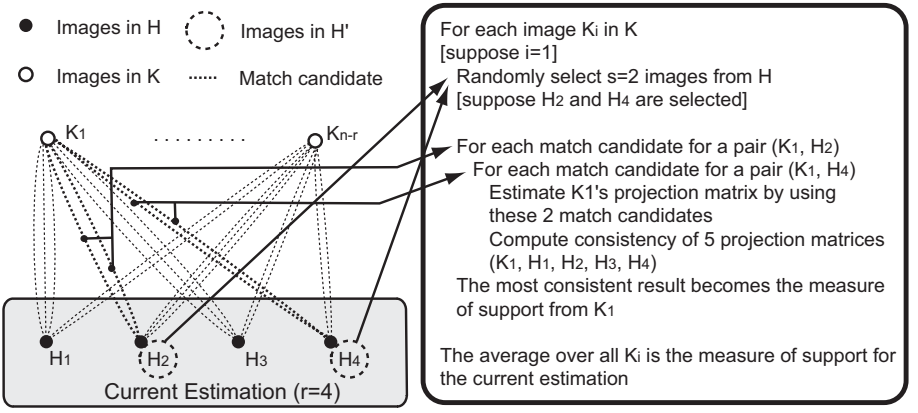



**Fig. 2.** A procedure for estimating how well $r$ projection matrices are supported by all the other images in the bootstrapping process.

4. *Smooth camera motion.* When the input images are part of a video sequence, it is possible to exploit the continuity of the camera motion. In particular, we require the angle between the viewing directions associated with images number $k$ and $l$ to be less than $|k-l|$ times some predefined threshold $d$. We use $d = 10$ [degrees] in our experiments.

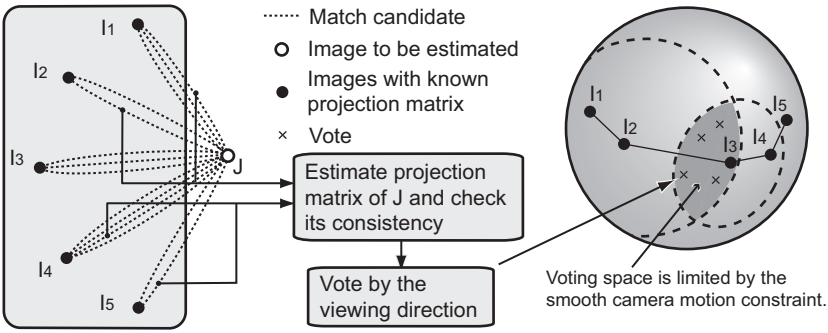



**Fig. 3.** Voting method to estimate a new projection matrix. Two match candidates are selected to cast a vote. When a camera motion is known to be smooth, the third consistency check method is applied and the voting space is limited to the intersection of circles.

**Selecting consistent matches while estimating motion parameters.** Let us show how to find geometrically consistent matches between image pairs while estimating the corresponding epipolar geometries as well as the individual projection matrices. As noted in Section 2.2, bootstrapping the process requires

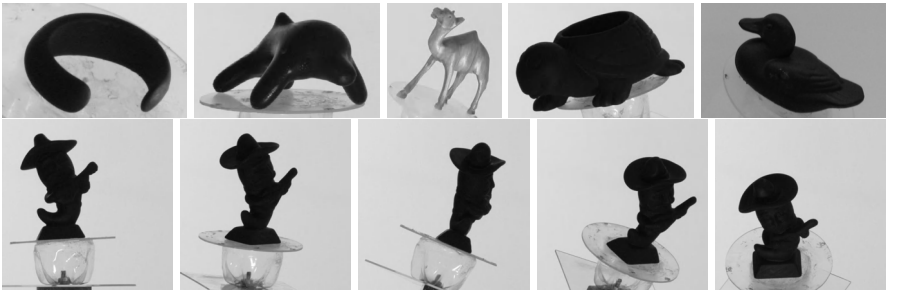


**Fig. 4.** Sample images of objects. The top row shows an image of a bracelet, a toy dolphin, a toy camel, a toy turtle, and a toy duck. The bottom row shows five images of a Mexican doll.

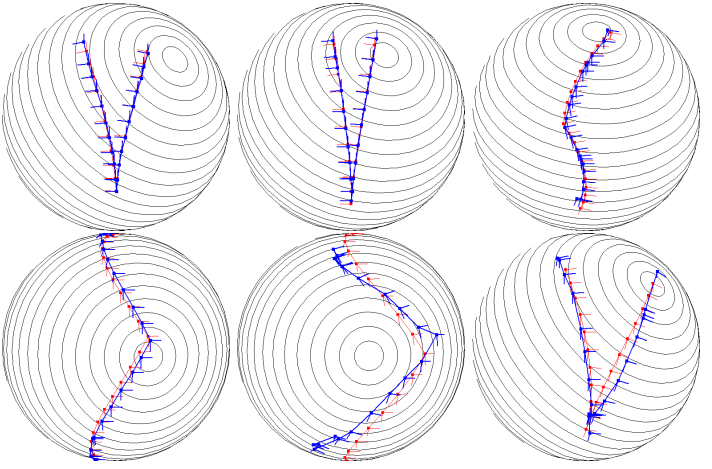


**Fig. 5.** In all the figures, thin lines represent ground truth data and thick lines represent our estimations. Top: recovered camera trajectories of bracelet, dolphin, and camel. Bottom: recovered camera trajectories of turtle, duck, and Mexican doll.

selecting $r \geq 3$ images from a total of $n$ images and one match candidate for each one of the $\binom{r}{2} \geq 3$ corresponding image pairs. First, we randomly select $r$ images $H = \{H_1, \ldots, H_r\}$ and try all promising matches among them to estimate $r$ projection matrices. Second, we measure how well these estimates are supported by the other images $K = \{K_1, \ldots, K_{n-r}\}$. After repeating this process a fixed number of times, we finally report the set $H$ of $r$ images with maximum support as the winner.

Our measure of support is defined as follows (Figure 2): Suppose for a moment that $\binom{r}{2}$ match candidates have been used to estimate the projection matrices associated with the $r$ images in $H$. For each image $K_i$ in $K$, $s \geq 2$ images are randomly selected from $H$ to estimate the projection matrix of $K_i$. Note that since the projection matrices associated with the elements of $H$ are known, we only need to match $K_i$ with $s \geq 2$ elements $H'$ of $H$ to estimate its projection

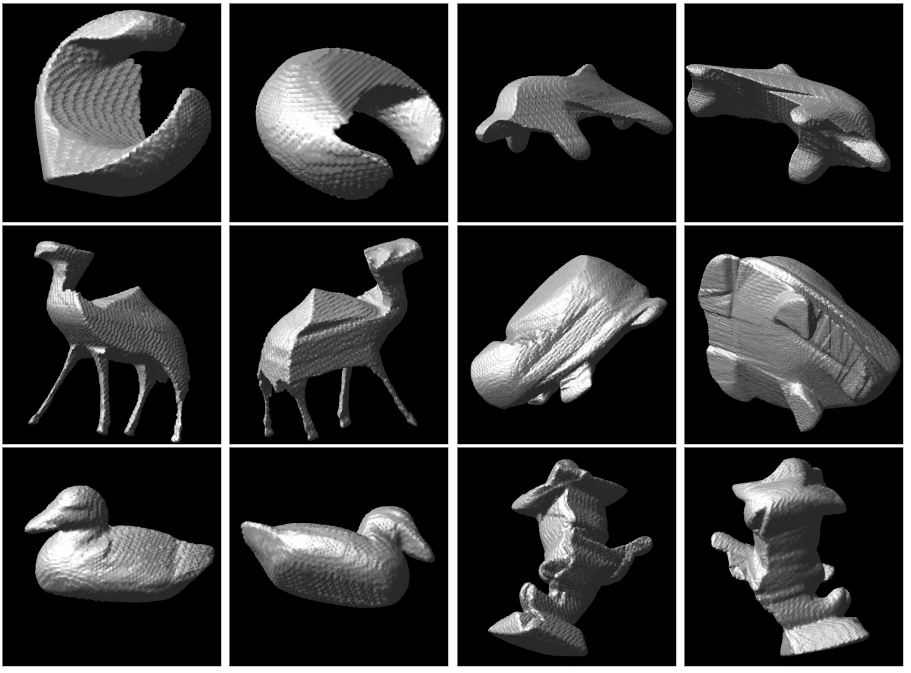
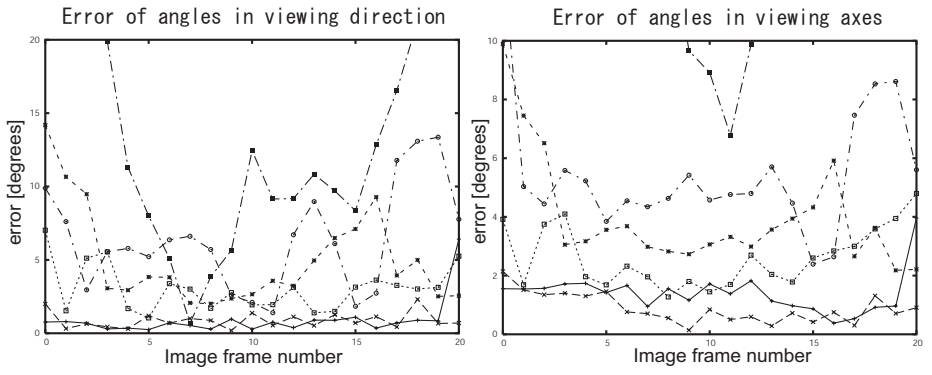


**Fig. 6.** Visual hull models constructed using the recovered camera projections.

matrix. For each image $K_i$ and each element of $H'$, we select one match candidate, estimate the projection matrix of $K_i$, and compute a consistency score by using the geometric constraints described above. This process is repeated for all tuples of match candidates between $K_i$ and $H'$, and we take the maximum consistency score as the measure of support $S(K_i)$ of the image $K_i$ for $H$. The overall measure of support for $H$ is computed as the average of the individual measures, or $\sum_{i=1}^{n-r} S(K_i)/(n-r)$. Next, we will describe how to estimate all the other $(n-r)$ projection matrices starting from the estimation of $r$ projection matrices that has been just computed.

Let us assume from now on that the projection matrices associated with $m \geq r$ images $I = \{I_1, \ldots, I_m\}$ have been computed, and consider the problem of adding one more image $J$ to $I$ (Figure 3). We use a voting scheme to improve the matching reliability: We tessellate the unit sphere and represent each projection matrix by its viewing direction on the sphere. For all tuples $I'$ of size $s$ of images in $I$ (again for the same reason as above, we need to match $J$ with only $s \geq 2$ other elements for the estimation), we exhaustively choose a match candidate between $J$ and each image in $I'$, then estimate the projection matrix for $J$. Its consistency is checked by enforcing the four geometric constraints given above, and we cast a vote. The cell receiving the largest number of votes is declared as a winner and simply an average is taken in that cell to estimate the projection matrix of $J$. Note that the motion smoothness constraint can be incorporated in this scheme by limiting the voting space as an intersection of circles, centered




Mean and standard deviation of error of angles

| Sequence | | bracelet | dolphin | camel | turtle | duck | mexican |
|---|---|---|---|---|---|---|---|
| Error in viewing direction [degrees] | Mean | 0.91 | 0.88 | 3.05 | 5.04 | 17.0 | 6.39 |
| | Standard Deviation | 1.26 | 0.53 | 1.53 | 3.24 | 14.0 | 3.46 |
| Error in viewing axes [degrees] | Mean | 1.40 | 0.89 | 2.61 | 3.98 | 26.0 | 5.50 |
| | Standard Deviation | 0.70 | 0.49 | 1.01 | 1.87 | 13.6 | 2.22 |

bracelet
dolphin
camel
turtle
duck
mexican

**Fig. 7.** Quantitative experimental results. Orientation errors in viewing directions and viewing axes are plotted for all the sequences. The mean and the standard deviation of these errors are also shown in the bottom table.

at viewing directions of each $I_i$, as shown in Figure 3. All images are added one by one to the set $I$ by using this simple voting strategy repeatedly.

## 3 Implementation Details and Experimental Results

Six objects (a bracelet, a toy dolphin, a toy camel, a toy turtle, a toy duck, and a Mexican doll) have been used in our experiments. Each sequence consists of 21 images, which are acquired using a pan-tilt head providing ground truth for the viewing angles. Figure 4 shows one sample image for the first five objects, and five images for the Mexican doll to illustrate its complex shape.

Image contours are extracted with sub-pixel localization using B-spline snakes and gradient vector flow [27], while detecting corners. As discussed in the previous section, our algorithm first finds a set of $r$ geometrically-consistent projection matrices by examining a subset of all the image tuples. The size of this subset has been set to 50 for all the examples. All other projection matrices are then estimated one by one. We exploit the smooth camera motion constraint for all the objects, using values of $r = 4$ and $s = 2$ in all cases.

Figure 5 compares the camera trajectories recovered by our algorithm to the ground-truth data from the pan-tilt head. In each case, the corresponding camera coordinate frames are first registered by a similarity transformation before being plotted on the unit sphere. As can be seen from the figure, estimated trajectories

are quite accurate, especially for the first four objects. As shown by Figure 6, the objects' visual hulls [3,16] are also recovered quite well. In fact, most inaccuracies are not so much due to errors in the recovered projection matrices as to the fact that a limited set of camera positions was used to construct each model.

Some quantitative results are given in Figure 7. The top two graphs show that errors tend to decrease in the middle of image sequences, which corresponds to intuition. As shown by the bottom table, rather large errors are obtained for the duck sequence. This is due do a few erroneous projection matrices at the beginning and the end of the sequence, with accurate estimates in its middle part.

## References


1. Amit Sethi, David Renaudie, David Kriegman, and Jean Ponce. Curve and Surface Duals and the Recognition of Curved 3D Objects from their Silhouette. *Int. J. of Comp. Vision*, 58(1), 2004.
2. Kalle Åström and Fredrik Kahl. Motion estimation in image sequences using the deformation of apparent contours. *IEEE Trans. Patt. Anal. Mach. Intell*, 21(2):114–127, 1999.
3. B.G. Baumgart. Geometric modeling for computer vision. Technical Report AIM-249, Stanford University, 1974. Ph.D. Thesis. Department of Computer Science.
4. S. Birchfield. KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker.
5. Edmond Boyer and Marie Odile Berger. 3d surface reconstruction using occluding contours. *Int. J. of Comp. Vision*, 22(3):219–233, 1997.
6. Roberto Cipolla, Kalle E. Åström, and Peter J. Giblin. Motion from the frontier of curved surfaces. In *Proc. Int. Conf. Comp. Vision*, pages 269–275, 1995.
7. Roberto Cipolla and Andrew Blake. Surface shape from the deformation of apparent contours. *Int. J. of Comp. Vision*, 9(2):83–112, 1992.
8. O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, 2001.
9. D.A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice-Hall, 2002.
10. P. Giblin and R Weiss. Epipolar curves on surfaces. *Image and Vision Computing*, 13(1):33–44, 1995.
11. Peter Giblin, Frank E. Pollick, and J. E. Rycroft. Recovery of an unknown axis of rotation from the profiles of a rotating surface. *Journal of Optical Society America*, pages 1976–1984, 1994.
12. Peter Giblin and Richard Weiss. Reconstruction of surface from profiles. In *Proc. Int. Conf. Comp. Vision*, pages 136–144, 1987.
13. R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2000.
14. Tanuja Joshi, Narendra Ahuja, and Jean Ponce. Structure and motion estimation from dynamic silhouettes under perspective projection. In *Proc. Int. Conf. Comp. Vision*, pages 290–295, 1995.
15. J.J. Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 13:321–330, 1984.
16. A. Laurentini. How far 3D shapes can be understood from 2D silhouettes. *IEEE Trans. Patt. Anal. Mach. Intell.*, 17(2):188–194, February 1995.

17. Noam Levi and Michael Werman. The viewing graph. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 518–522, 2003.
18. Paulo Mendonca, Kwan-Yee K. Wong, and Robert Cipolla. Camera pose estimation and reconstruction from image profiles under circular motion. In *Proc. Euro. Conf. Comp. Vision*, pages 864–877, 2000.
19. C.J. Poelman and T. Kanade. A paraperspective factorization method for shape and motion recovery. *IEEE Trans. Patt. Anal. Mach. Intell.*, 19(3):206–218, March 1997.
20. Richard Szeliski and Richard Weiss. Robust shape recovery from occluding contours using a linear smoother. *Int. J. of Comp. Vision*, 28(1):27–44, 1998.
21. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. of Comp. Vision*, 9(2):137–154, 1992.
22. P. Torr and D. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. J. of Comp. Vision*, 24(3), 1997.
23. P.H. Torr, A. Zisserman, and S.J. Maybank. Robust detection of degenerate configurations for the fundamental matrix. In *Proc. Int. Conf. Comp. Vision*, pages 1037–1042, Boston, MA, 1995.
24. P.H.S. Torr and A. Zisserman. Mlesac: A new robust estimator with application to estimating image geometry. *CVIU*, 78(1):138–156, 2000.
25. Régis Vaillant and Olivier D. Faugeras. Using extremal boundaries for 3-d object modeling. *IEEE Trans. Patt. Anal. Mach. Intell*, 14(2):157–173, 1992.
26. B. Vijayakumar, David J. Kriegman, and Jean Ponce. Structure and motion of curved 3d objects from monocular silhouettes. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 327–334, 1996.
27. Yue Wang, Eam Khwang Teoh, and Dinggang Shen. Structure-adaptive b-snake for segmenting complex objects. In *IEEE International Conference on Image Processing*, 2001.
28. Kwan-Yee K. Wong and Robert Cipolla. Structure and motion from silhouettes. In *Proc. Int. Conf. Comp. Vision*, pages 217–222, 2001.

# Automatic Non-rigid 3D Modeling from Video

Lorenzo Torresani[1] and Aaron Hertzmann[2]

[1] Stanford University, Stanford, CA, USA
ltorresa@cs.stanford.edu
[2] University of Toronto, Toronto, ON, Canada
hertzman@dgp.toronto.edu

**Abstract.** We present a robust framework for estimating non-rigid 3D shape and motion in video sequences. Given an input video sequence, and a user-specified region to reconstruct, the algorithm automatically solves for the 3D time-varying shape and motion of the object, and estimates which pixels are outliers, while learning all system parameters, including a PDF over non-rigid deformations. There are no user-tuned parameters (other than initialization); all parameters are learned by maximizing the likelihood of the entire image stream. We apply our method to both rigid and non-rigid shape reconstruction, and demonstrate it in challenging cases of occlusion and variable illumination.

## 1 Introduction

Reconstruction from video promises to produce high-quality 3D models for many applications, such as video analysis and computer animation. Recenly, several "direct" methods for shape reconstruction from video sequences have been demonstrated [1,2,3] that can give good 3D reconstructions of non-rigid 3D shape, even from single-view video. Such methods estimate shape by direct optimization with respect to raw image data, thus avoiding the difficult problem of tracking features in advance of reconstruction. However, many difficulties remain for developing general-purpose video-based reconstruction algorithms. First, existing algorithms make the restrictive assumption of *color constancy*, that object features appear the same in all views. Almost all sequences of interest violate this assumption at times, such as with occlusions, lighting changes, motion blur, and many other common effects. Second, non-rigid shape reconstruction requires a number of regularization parameters (or, equivalently, prior distributions), due to fundamental ambiguities in non-rigid reconstruction [4,5], and to handle noise and prevent over-fitting. Such weights must either be tuned by hand (which is difficult and inaccurate, especially for models with many parameters) or learned from annotated training data (which is often unavailable, or inappropriate to the target data).

In this paper, we describe an algorithm for robust non-rigid shape reconstruction from uncalibrated video. Our general approach is to pose shape reconstruction as a maximum likelihood estimation problem, to be optimized with respect to the entire input video sequence. We solve for 3D time-varying

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 299–312, 2004.
© Springer-Verlag Berlin Heidelberg 2004

shape, correspondence, and outlier pixels, while *simultaneously* solving for all weighting/PDF parameters. By doing so, we exploit the general property of Bayesian learning that all parameters may be estimated by maximizing the posterior distribution for a suitable model. No prior training data or parameter tuning is required. This general methodology — of simultaneously solving for shape while learning all weights/PDF parameters — has not been applied to 3D shape reconstruction from video, and has only rarely been exploited in computer vision in general (one example is [6]).

This paper begins with a general framework for robust shape reconstruction from video. This model is based on robust statistics: all violations of color constancy are modeled as outliers. Unlike robust tracking algorithms, we solve for shape globally over an entire sequence, allowing us to handle cases where many features are completely occluded in some frames. (A disadvantage of our approach is that it cannot currently be applied to one-frame-at-a-time tracking). We demonstrate sequences for which previous global reconstruction methods fail. For example, previous direct methods require that all feature points be visible in all video frames, i.e. all features are visible in a single "reference frame;" our method relaxes this assumption and allows sequences for which no single "reference frame" exists. We also show examples where existing techniques fail due to local changes in lighting and shape. Our method is based on the EM algorithm for robust outlier detection. Additionally, we show how to simultaneously solve for the outlier probabilities for the target sequence.

We demonstrate the reconstruction framework in the case of rigid motion under weak perspective projection, and non-rigid shape under orthographic projection. In the latter case, we do not assume that the non-rigid geometry is known in advance. Separating non-rigid deformation from rigid motion is ambiguous without some assumptions about deformation [4,5]. Rather than specify the parameters of a shape prior in advance, our algorithm learns a shape PDF simultaneously with estimating correspondence, 3D shape reconstruction, and outliers.

### 1.1 Relation to Previous Work

We build on recent techniques for exploiting rank constraints on optical flow in uncalibrated single-view video. Conventional optical flow algorithms use only local information; namely, every track in every frame is estimated separately [7,8]. In contrast, so-called "direct methods" optimize directly with respect to the raw image sequence [9]. Irani [1] treated optical flow in rigid 3D scenes as a global problem, combining information from the entire sequence — along with rank constraints on motion — to yield better tracking. Bregler et al. [10] describe an algorithm for solving for non-rigid 3D shape from known point tracks. Extending these ideas, Torresani et al. [2,11] and Brand [3] describe tracking and reconstruction algorithms that solve for 3D shape and motion from sequences, even for non-rigid scenes. Note that adding robustness to the above methods is non-trivial, since this would require defining a unified objective function for tracking and reconstruction that is not present in the previous work. Furthermore, one

must introduce a large number of hand-tuned weighting and regularization constraints, especially for non-rigid motion, for which reconstruction is ill-posed without some form of regularization [4,5]. In our paper, we show how to cast the problem of estimating 3D shape and motion from video sequences as optimization of a well-defined likelihood function. This framework allows several extensions: our method automatically detects outliers, and all regularization parameters are automatically learned via Bayesian learning. Our non-rigid model incorporates our previous work on non-rigid structure-from-motion [5], in which reliable tracking data was assumed to be available in advance.

A common approach to acquiring rigid shape from video is to separate feature selection, outlier rejection, and shape reconstruction into a series of stages, each of which has a separate optimization process (e.g. [12]). Dellaert et al. [13] solve for rigid shape while detecting outliers, assuming that good features can be located in advance. The above methods assume that good features can be detected in each frame by a feature detector, and that noise/outlier parameters are known in advance. In contrast to these methods, we optimize the reconstruction directly with respect to the video sequence.

Robust algorithms for tracking have been widely explored in local tracking (e.g. [14,15,16]). Unlike local robust methods, our method can handle features that are completely occluded, by making use of global constraints on motion. Similar to Jepson et al. [16], we also learn the parameters at the same time as tracking, rather than assuming that they are known *a priori*. Our outlier model is closely related to layer-based motion segmentation algorithms [6,17,18], which are also often applied globally to a sequence. We use the outlier model to handle general violations of color constancy, rather than to specifically model multiple layers.

## 2 Robust Shape Reconstruction Framework

We now describe our general framework for robust shape reconstruction from uncalibrated video. We then specialize this framework to rigid 3D motion in Section 3, and to non-rigid motion in Section 4.

### 2.1 Motion Model

We assume that 3D shape can be described in terms of the 3D coordinates $\mathbf{s}_{j,t} = [X_{j,t}, Y_{j,t}, Z_{j,t}]^T$ of $J$ scene points, over $T$ time steps. The parameter $j$ indexes over points in the model, and $t$ over time. We collect these points in a matrix $\mathbf{S}_t = [\mathbf{s}_{1,t}, ..., \mathbf{s}_{J,t}]$. We parameterize 3D shape with a function as $\mathbf{S}_t = \Gamma(\mathbf{z}_t; \psi)$, where $\mathbf{z}_t$ is a hidden random variable describing the shape at each time $t$, with a prior $p(\mathbf{z}_t)$; $\psi$ are shape model parameters. The details of these functions depend on the application. For example, in the case of rigid shape (Section 3), we use $\Gamma(\mathbf{z}_t; \psi) = \bar{\mathbf{S}}$, i.e. the shape stays fixed at a constant value $\bar{\mathbf{S}}$, and $\psi = \{\bar{\mathbf{S}}\}$. For non-rigid shape (Section 4), $\Gamma(\mathbf{z}_t; \psi)$ is a linear combination

of basis shapes, determined by the time-varying weights $\mathbf{z}_t$; $\psi$ contains the shape basis.

Additionally, we define a camera model $\Pi$. At a given time $t$, point $j$ projects to a 2D position $\mathbf{p}_{j,t} = [x_{j,t}, y_{j,t}]^T = \Pi(\mathbf{s}_{j,t}; \xi_t)$, where $\xi_t$ are the time-varying parameters of the camera model. For example, $\xi_t$ might define the position and orientation of the camera with respect to the object.

In cases when the object is undergoing rigid motion, we subsume it in the rigid motion of the camera. This applies in both the case of rigid shape and non-rigid shape. In the non-rigid case, we can generally think of the object's motion as consisting of a rigid component plus a non-rigid deformation. For example, a person's head can move rigidly (e.g. turning left or right) while deforming (due to changing facial expressions). One might wish to separate rigid object motion from rigid camera motion in other applications, such as under perspective projection.

### 2.2 Image Model

We now introduce a generative model for video sequences, given the motion of the 2D point tracks $\mathbf{p}_{j,t}$. Individual images in a video sequence are created from the 2D points. Ideally, the window of pixels around each point $\mathbf{p}_{j,t}$ should remain constant over time; however, this window may be corrupted by noise and outliers. Let $w$ be an index over a pixel window, so that $I_w(\mathbf{p}_{j,t})$ is the intensity of a pixel in the window[1] of point $j$ in frame $t$. This pixel intensity should ideally be a constant $\bar{I}_{w,j}$; however, it will be corrupted by Gaussian noise with variance $\sigma^2$. Moreover, it may be replaced by an outlier, with probability $1-\tau$. We define a hidden variable $W_{w,j,t}$ so that $W_{w,j,t} = 0$ if the pixel is replaced by an outlier, and $W_{w,j,t} = 1$ if it is valid. The complete PDF over individual pixels in a window is given by:

$$p(W_{w,j,t} = 1) = \tau \tag{1}$$

$$p(I_w(\mathbf{p}_{j,t}) | W_{w,j,t} = 1, \mathbf{p}_{j,t}, \bar{I}_{w,j}, \sigma^2) = \mathcal{N}(I_w(\mathbf{p}_{j,t}) | \bar{I}_{w,j}; \sigma^2) \tag{2}$$

$$p(I_w(\mathbf{p}_{j,t}) | W_{w,j,t} = 0, \mathbf{p}_{j,t}, \bar{I}_{w,j}, \sigma^2) = c \tag{3}$$

where $\mathcal{N}(I_w(\mathbf{p}_{j,t}) | \bar{I}_{w,j}; \sigma^2)$ denotes a 1D Gaussian distribution with mean $\bar{I}_{w,j}$ and variance $\sigma^2$, and $c$ is a constant corresponding to the uniform distribution over the range of valid pixel intensities. The values $\bar{I}_{w,j}$ are determined by the corresponding pixel in the reference frame.

For convenience, we do not model the appearance of video pixels that do not appear near 2D points, or correlations between pixels when windows overlap.

### 2.3 Problem Statement

Given a video sequence $I$ and 2D point positions specified in some reference frames, we would like to estimate the positions of the points in all other frames, and, additionally, learn the 3D shape and associated parameters.

[1] In other words, $I_w(\mathbf{p}_{j,t}) = I^{(t)}(\mathbf{p}_{j,t} + \mathbf{d}_w)$ where $I^{(t)}$ is the image at time $t$, and $\mathbf{d}_w$ represents the offset of point $w$ inside the window.

We propose to solve this estimation problem by maximizing the likelihood of the image sequence given the model. We encapsulate the parameters for the image, shape, and camera model into the parameter vector $\theta = \{\bar{I}, \sigma^2, \tau, \psi, \xi_1, ..., \xi_T\}$. The likelihood itself marginalizes over the hidden variables $W_{w,j,t}$ and $\mathbf{z}_t$. Consequently, our goal is to solve for $\theta$ to maximize

$$p(I|\theta) = \prod_{w,j,t} p(I_w(\mathbf{p}_{j,t})|\theta) = \prod_{w,j,t} \int_{\mathbf{z}_t} \sum_{W_{w,j,t} \in \{0,1\}} p(I, \mathbf{z}_t, W_{w,j,t}|\theta) d\mathbf{z}_t \quad (4)$$

(We have replaced $I_w(\mathbf{p}_{j,t})$ with $I$ for brevity). In other words, we wish to solve for the camera motion, shape PDF, and outlier distribution from the video sequence $I$, averaging over the unknown shapes and outliers.

### 2.4 Variational Bound

In order to optimize Equation 4, we use an approach based on variational learning [19]. Specifically, we introduce a distribution $Q(W_{w,j,t}, \mathbf{z}_t)$ to approximate the distribution over the hidden parameters at time $t$, and then apply Jensen's inequality to derive an upper bound on the negative log likelihood:[2]

$$-\ln p(I|\theta) = -\ln \prod_{w,j,t} \int_{\mathbf{z}_t} \sum_{W_{w,j,t} \in \{0,1\}} p(I, \mathbf{z}_t, W_{w,j,t}|\theta) d\mathbf{z}_t \quad (5)$$

$$= -\sum_{w,j,t} \ln \int_{\mathbf{z}_t} \sum_{W_{w,j,t} \in \{0,1\}} p(I, \mathbf{z}_t, W_{w,j,t}|\theta) \frac{Q(W_{w,j,t}, \mathbf{z}_t)}{Q(W_{w,j,t}, \mathbf{z}_t)} d\mathbf{z}_t \quad (6)$$

$$\leq -\sum_{w,j,t} \int_{\mathbf{z}_t} \sum_{W_{w,j,t} \in \{0,1\}} Q(W_{w,j,t}, \mathbf{z}_t) \ln \frac{p(I, \mathbf{z}_t, W_{w,j,t}|\theta)}{Q(W_{w,j,t}, \mathbf{z}_t)} d\mathbf{z}_t \quad (7)$$

We can minimize the negative log likelihood by minimizing Equation 7 with respect to $\theta$ and $Q$. Unfortunately, even representing the optimal distribution $Q(W_{w,j,t}, \mathbf{z}_t)$ would be intractable, due to the large number of point tracks. To make it manageable, we represent the distribution $Q$ with a factored form: $Q(W_{w,j,t}, \mathbf{z}_t) = q(\mathbf{z}_t)q(W_{w,j,t})$, where $q(\mathbf{z}_t)$ is a distribution over $\mathbf{z}_t$ for each frame, and $q(W_{w,j,t})$ is a distribution over whether each pixel $(w, j, t)$ is valid or an outlier. The distribution $q(\mathbf{z}_t)$ can be thought of as approximating to $p(\mathbf{z}_t|I, \theta)$, and $q(W_{w,j,t})$ approximates the distribution $p(W_{w,j,t}|I, \theta)$. Substituting the factored form into Equation 7 gives the variational free energy (VFE) $\mathcal{F}(\theta, q)$:

$$\mathcal{F}(\theta, q) = -\sum_{w,j,t} \int_{\mathbf{z}_t} \sum_{W_{w,j,t} \in \{0,1\}} q(W_{w,j,t}) q(\mathbf{z}_t) \ln \frac{p(I, \mathbf{z}_t, W_{w,j,t}|\theta)}{q(W_{w,j,t}) q(\mathbf{z}_t)} d\mathbf{z}_t \quad (8)$$

In order to estimate shape and motion from video, our new goal is to minimize $\mathcal{F}$ with respect to $\theta$ and $q$ over all points $j$ and frames $t$. For brevity, we write

[2] We require that $\prod_{w,j,t} \int_{\mathbf{z}_t} \sum_{W_{w,j,t} \in \{0,1\}} Q(W_{w,j,t}, \mathbf{z}_t) = 1$, in order for Jensen's inequality to hold.

$\gamma_{w,j,t} \equiv q(W_{w,j,t} = 1)$. Substituting the image model from Section 2.2 and defining the expectation $E_{q(\mathbf{z}_t)}[f(\mathbf{z}_t)] \equiv \int q(\mathbf{z}_t) f(\mathbf{z}_t) d\mathbf{z}_t$ gives

$$\begin{aligned}\mathcal{F}(\theta, q, \gamma) = &\sum_{w,j,t} \gamma_{w,j,t} E_{q(\mathbf{z}_t)}[(I_w(\mathbf{p}_{t,j}) - \bar{I}_{w,j})^2]/(2\sigma^2) + \ln\sqrt{2\pi\sigma^2} \sum_{w,j,t} \gamma_{w,j,t} \\ &-NJ \sum_t E_{q(\mathbf{z}_t)}[\ln p(\mathbf{z}_t)] - \ln c \sum_{w,j,t} (1 - \gamma_{w,j,t}) - \ln\tau \sum_{w,j,t} \gamma_{w,j,t} \\ &- \ln(1-\tau) \sum_{w,j,t} (1 - \gamma_{w,j,t}) + NJ \sum_t E_{q(\mathbf{z}_t)}[\ln q(\mathbf{z}_t)] + \\ &\sum_{w,j,t} (1 - \gamma_{w,j,t}) \ln(1 - \gamma_{w,j,t}) + \sum_{w,j,t} \gamma_{w,j,t} \ln \gamma_{w,j,t} + \textit{constants} \quad (9)\end{aligned}$$

where $N$ is the number of pixels in a window. Although there are many terms in this expression, most terms have a simple interpretation. Specifically, we point out that the first term is a weighted image matching term: for each pixel, it measures the expected reconstruction error from comparing an image pixel to its mean intensity $\bar{I}_{w,j}$, weighted by the likelihood $\gamma_{w,j,t}$ that the pixel is valid.

### 2.5 Generalized EM Algorithm

We optimize the VFE using a generalized EM algorithm. In the E-step we keep the model parameters fixed and update our estimate of the hidden variable distributions. The update rule for $q(\mathbf{z}_t)$ will depend on the particular motion model specified by $\Gamma$. The distribution $\gamma_{w,j,t}$ (which indicates whether pixel $(w, j, t)$ is an outlier) is estimated as:

$$\alpha_0 = p(I_w(\mathbf{p}_{j,t}) | W_{w,j,t} = 0, \mathbf{p}_{j,t}, \theta) p(W_{w,j,t} = 0 | \theta) = (1 - \tau)c \quad (10)$$

$$\alpha_1 = p(I_w(\mathbf{p}_{j,t}) | W_{w,j,t} = 1, \mathbf{p}_{j,t}, \theta) p(W_{w,j,t} = 1 | \theta) \quad (11)$$

$$= \frac{\tau}{\sqrt{2\pi\sigma^2}} e^{-E_{q(\mathbf{z}_t)}[(I_w(\mathbf{p}_{t,j}) - \bar{I}_{w,j})^2]/(2\sigma^2)} \quad (12)$$

Then, using Bayes' Rule, we have the E-step for $\gamma_{w,j,t}$:

$$\gamma_{w,j,t} \leftarrow \alpha_1/(\alpha_0 + \alpha_1) \quad (13)$$

In the generalized M-step, we solve for optical flow and 3D shape given the outlier probabilities $\gamma_{w,j,t}$. The outlier probabilities provide a weighting function for tracking and reconstruction: pixels likely to be valid are given more weight. Let $\mathbf{p}^0_{j,t}$ represent the current estimate of $\mathbf{p}_{j,t}$ at a step during the optimization. To solve for the motion parameters that define $\mathbf{p}_{j,t}$, we linearize the target image around $\mathbf{p}^0_{j,t}$:

$$I_w(\mathbf{p}_{j,t}) \approx I_w(\mathbf{p}^0_{j,t}) + \nabla I_w^T (\mathbf{p}_{j,t} - \mathbf{p}^0_{j,t}) \quad (14)$$

where $\nabla I_w$ denotes a 2D vector of image derivatives at $I_w(\mathbf{p}^0_{j,t})$. One such linearization is applied for every pixel $w$ in every window $j$ for every frame $t$ at every iteration of the algorithm.

Substituting Equation 14 into the first term of the VFE (Equation 9) yields the following quadratic energy function for the motion:[3]

$$\sum_{w,j,t} \frac{\gamma_{w,j,t}}{2\sigma^2} E_{q(\mathbf{z}_t)}[(I_w(\mathbf{p}_{t,j}) - \bar{I}_{w,j})^2] \approx \sum_{j,t} E_{q(\mathbf{z}_t)}[(\mathbf{p}_{j,t} - \hat{\mathbf{p}}_{j,t})^T \mathbf{e}_{j,t} (\mathbf{p}_{j,t} - \hat{\mathbf{p}}_{j,t})] \tag{15}$$

where

$$\mathbf{e}_{j,t} = \frac{1}{2\sigma^2} \sum_w \gamma_{w,j,t} \nabla I_w \nabla I_w^T \tag{16}$$

$$\hat{\mathbf{p}}_{j,t} = \mathbf{p}_{j,t}^0 + \frac{1}{2\sigma^2} \mathbf{e}_{j,t}^{-1} \sum_w \gamma_{w,j,t} (\bar{I}_{w,j} - I_w(\mathbf{p}_{j,t}^0)) \nabla I_w \tag{17}$$

Hence, optimizing the shape and motion with respect to the image is equivalent to solving the structure-from-motion problem of fitting the "virtual point tracks" $\hat{\mathbf{p}}_{j,t}$, each of which has uncertainty specified by a $2 \times 2$ covariance matrix $\mathbf{e}_{j,t}^{-1}$. In the next sections, we will outline the details of this optimization for both rigid and non-rigid motion.

The noise variance and the outlier prior probability are also updated in the M-step, by optimizing $\mathcal{F}(\theta, q, \gamma)$ for $\tau$ and $\sigma^2$:

$$\tau \leftarrow \sum_{w,j} \gamma_{w,j,t} / (JNT) \tag{18}$$

$$\sigma^2 \leftarrow \sum_{w,j,t} \gamma_{w,j,t} E_{q(\mathbf{z}_t)}[(I_w(\mathbf{p}_{j,t}) - \bar{I}_{w,j})^2] / \sum_{w,j} \gamma_{w,j,t} \tag{19}$$

The $\sigma^2$ update can be computed with Equation 15. These updates can be interpreted as the expected percentage of outliers, and the expected image variance, respectively.

## 3 Rigid 3D Shape Reconstruction from Video

The general-purpose framework presented in the previous section can be specialized to a variety of projection and motion models. In this section we outline the algorithm in the case of rigid motion under weak orthographic projection.

This projection model can be described in terms of parameters $\xi_t = \{\alpha_t, \mathbf{R}_t, \mathbf{t}_t\}$ and projection function

$$\mathit{\Pi}(\mathbf{s}_{j,t}; \{\alpha_t, \mathbf{R}_t, \mathbf{t}_t\}) = \alpha_t \mathbf{R}_t \mathbf{s}_{j,t} + \mathbf{t}_t \tag{20}$$

where $\mathbf{R}_t$ is a $2 \times 3$ matrix combining rotation with orthographic projection, $\mathbf{t}_t$ is a $2 \times 1$ translation vector and $\alpha_t$ is a scalar implicitly representing the weak perspective scaling $(f/Z_{avg})$. The 3D shape of the object is assumed to

[3] The linearized VFE is not guaranteed to bound the negative log-likelihood, but provides a local approximation to the actual VFE.

remain constant over the entire sequence and, thus, we can use as our shape model $\Gamma(\psi) = \bar{\mathbf{S}}$, without introducing a time-dependent latent variable $\mathbf{z}_t$. In other words, the model for 2D points is $\mathbf{p}_{j,t} = \alpha_t \mathbf{R}_t \bar{\mathbf{s}}_{j,t} + \mathbf{t}_t$.

For this case, the objective function in Equation 9 reduces to:

$$\begin{aligned}\mathcal{F}(\theta, \mathbf{R}, \mathbf{t}, \gamma) = &\sum_{w,j,t} \gamma_{w,j,t} (I_w(\mathbf{p}_{t,j}) - \bar{I}_{w,j})^2/(2\sigma^2) + \sum_{w,j,t} \gamma_{w,j,t} \ln \sqrt{2\pi\sigma^2} \\ &- \sum_{w,j,t} \gamma_{w,j,t} \ln \tau - \sum_{w,j,t} (1 - \gamma_{w,j,t}) \ln c(1-\tau) \\ &+ \sum_{w,j,t} \gamma_{w,j,t} \ln \gamma_{w,j,t} + \sum_{w,j,t} (1 - \gamma_{w,j,t}) \ln(1 - \gamma_{w,j,t}) \end{aligned} \quad (21)$$

Note that, in the case where all pixels are completely reliable (all $\gamma_{w,j,t} = 1$), this reduces to a global image matching objective function. Again, we can rewrite the first term in the free energy in terms of virtual point tracks $\hat{\mathbf{p}}_{j,t}$ and covariances, as in Equation 15. Covariance-weighted factorization [20] can then be applied to minimize this objective function to estimate the rigid shape $\bar{\mathbf{S}}$ and the motion parameters $\mathbf{R}_t$, $\mathbf{t}_t$ and $\alpha_t$ for all frames. Orthonormality constraints on rotation matrices are enforced in a fashion similar to [21]. To summarize the entire algorithm, we alternate between optimizing each of $\gamma_{w,j,t}$, $\mathbf{R}_t$, $\mathbf{t}_t$, $\alpha_t$, $\tau$, and $\sigma^2$. Between each of the updates, $\hat{\mathbf{p}}_{j,t}$ and $\mathbf{e}_{j,t}$ are recomputed.

*Implementation details.* We initialize our algorithm using conventional coarse-to-fine Lucas-Kanade tracking [8]. Since the conventional tracker will diverge if applied to the entire sequence at once, we correct the motion every few frames by applying our generalized EM algorithm over the subsequence thus far initialized. This process is repeated until we reach the end of the sequence. We refine this estimate by additional EM iterations. The values of $\sigma^2$ and $\tau$ are initially held fixed at 10 and 0.3, respectively. They are then updated in every M-step after the first few iterations.

*Experiments.* We applied the robust reconstruction algorithm to a sequence assuming rigid motion under weak perspective projection. This video contains 100 frames of mostly-rigid head/face motion. The sequence is challenging due to the low resolution and low frame rate (15 fps). In this example, there is no single frame in which feature points from both sides of the face are clearly visible, so existing global techniques cannot be applied.

To test our algorithm, we manually indicated regions-of-interest in two reference frames, from which 45 features were automatically selected (Figure 1(a)). Points from the left side of the subject's face are occluded for more than 50% of the sequence. Some of the features on the left side of the face are lost or incorrectly tracked by local methods after just four frames from the reference image where they were selected. Within 14 frames, all points from the left side are completely invisible, and thus would be lost by conventional techniques. With robust reconstruction, our algorithm successfully tracks all features, making use of learned geometry constraints to fill in missing features (Figure 2).

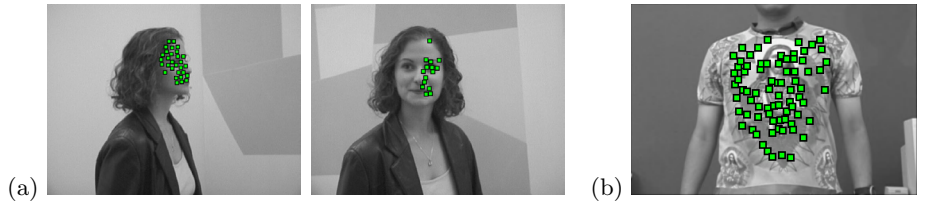



**Fig. 1.** Reference frames. Regions of interest were selected manually, and individual point locations selected automatically using Shi and Tomasi's method [22]. Note that, in the first sequence, most points are clearly visible in only one reference frame. (Refer to the electronic version of this paper to view the points in color.)

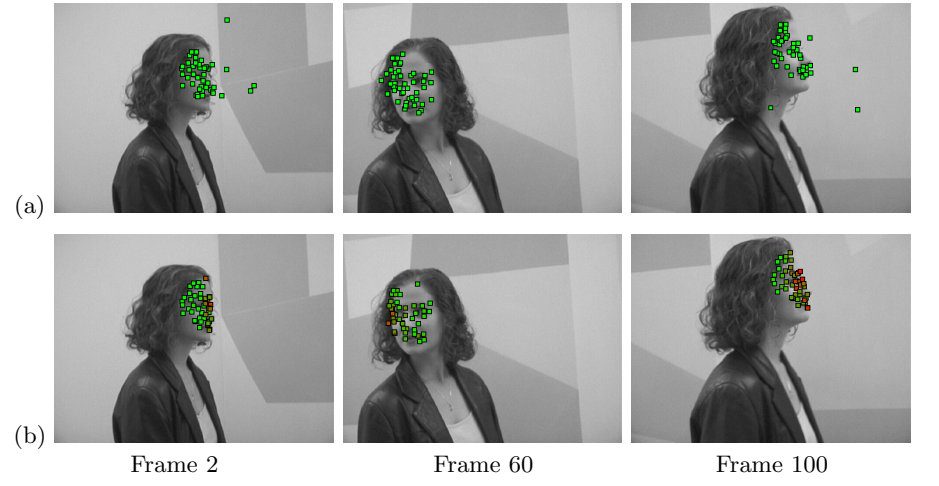



**Fig. 2.** (a) Rank-constrained tracking of the rigid sequence without outlier detection (i.e. using $\tau = 0$), using the reference frames shown in Figure 1(a). Tracks on occluded portions of the face are consistently lost. (b) Robust, rank-constrained tracking applied to the same sequence. Tracks are colored according to the average value of $\gamma_{w,j,t}$ for the pixels in the track's window: green for completely valid pixels, and red for all outliers.

## 4 Non-rigid 3D Shape Reconstruction from Video

We now apply our framework to the case where 3D shape consists of both rigid motion and non-rigid deformation, and show how to solve for the deforming shape from video, while detecting outliers and solving for the shape and outlier PDFs. Our approach builds on our previous algorithm for non-rigid structure-from-motion [5], which, as previously demonstrated on toy examples, yields much better reconstructions than applying a user-defined regularization.

We assume that the nonrigid shape $\mathbf{S}_t$ at time $t$ can be described as a "shape average" $\bar{\mathbf{S}}$ plus a linear combination of $K$ basis shapes $\mathbf{V}_k$:

$$\mathbf{S}_t = \Gamma(\mathbf{z}_t; \psi) = \bar{\mathbf{S}} + \sum_{k=1}^{K} \mathbf{V}_k z_{k,t} \tag{22}$$

where $k$ indexes elements of $\mathbf{z}_t$ and $\psi = \{\bar{\mathbf{S}}, \mathbf{V}_1, ..., \mathbf{V}_K\}$. The scalar weights $z_{k,t}$ indicate the deformation in each frame $t$. Together, $\bar{\mathbf{S}}$ and $\mathbf{V}_k$ are referred to as the *shape basis*. The $\mathbf{z}_t$ are Gaussian hidden variables with zero mean and unit variance ($p(\mathbf{z}_t) = \mathcal{N}(\mathbf{z}_t|0; \mathbf{I})$). With $\mathbf{z}_t$ treated as a hidden variable, this model is a factor analyzer, and the distribution over shape $p(\mathbf{S}_t)$ is Gaussian. See [5] for a more detailed discussion of this model. Scene points are viewed under orthographic projection according to the model: $\Pi(\mathbf{s}_{j,t}; \{\mathbf{R}_t, \mathbf{t}_t\}) = \mathbf{R}_t \mathbf{s}_{j,t} + \mathbf{t}_t$. The imaging model is the same as described in Section 2.2. We encapsulate the model in the parameter vector $\theta = \{\bar{I}, \sigma^2, \tau, \mathbf{R}_1, ..., \mathbf{R}_T, \mathbf{t}_1, ..., \mathbf{t}_T, \bar{\mathbf{S}}, \mathbf{V}_1, ..., \mathbf{V}_K\}$.

We optimize the VFE by alternating updates of each of the parameters. Each update entails setting $\partial\mathcal{F}$ to zero with respect to each of the parameters; e.g. $\mathbf{t}_t$ is updated by solving $\frac{\partial \mathcal{F}}{\partial \mathbf{t}_t} = 0$. The algorithm is given in the appendix.

*Experiments.* We tested our integrated 3D reconstruction algorithm on a challenging video sequence of non-rigid human motion. The video consists of 660 frames recorded in our lab with a consumer digital video camera and contains non-rigid deformations of a human torso. Although most of the features tracked are characterized by distinctive 2D texture, their local appearance changes considerably during the sequence due to occlusions, shape deformations, varying illumination in patches, and motion blur. More than 25% of the frames contain occluded features, due to arm motion and large torso rotations. 77 features were selected automatically in the first frame using the criterion described by Shi and Tomasi [22]. Figure 1(b) shows their initial locations in the reference frame. The sequence was initially processed assuming $K = 1$ (corresponding to rigid motion plus a single mode of deformation), and increased to $K = 2$ during optimization. Estimated positions of features with and without robustness are shown in Figure 3. As shown in Figure 3(a), tracking without outlier detection fails to converge to a reasonable result, even if initialized with the results of the robust algorithm. 3D reconstructions from our algorithm are shown in Figure 4(b). The resulting 3D shape is highly detailed, even for occluded regions. For comparison, we applied robust rank-constrained tracking to solve for maximum likelihood $\mathbf{z}_t$ and $\theta$, followed by applying the EM-Gaussian algorithm [5] to the recovered point tracks. Although the results are mostly reasonable, a few significant errors occur in an occluded region. Our algorithm avoids these errors, because it optimizes all parameters directly with respect to the raw image data. Additional results and visualizations are shown at `http://movement.stanford.edu/automatic-nr-modeling/`

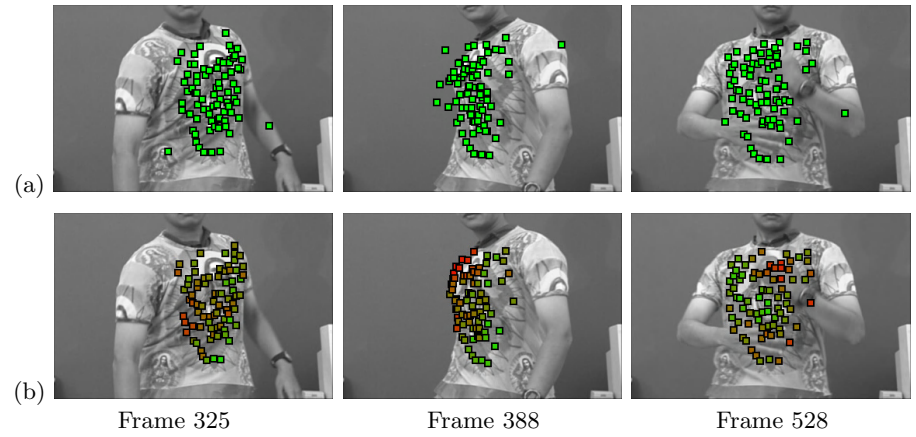



**Fig. 3.** (a) Rank-constrained tracking of the second sequence without outlier detection fails to converge to a reasonable result. Here we show that, even when initialized with the solution from the robust method, tracking without robustness causes the results to degrade. (b) Robust, rank-constrained tracking applied to the same sequence. Tracks are colored according to the average value of $\gamma_{w,j,t}$ for the pixels in the track's window: green for completely valid pixels, and red for all outliers.

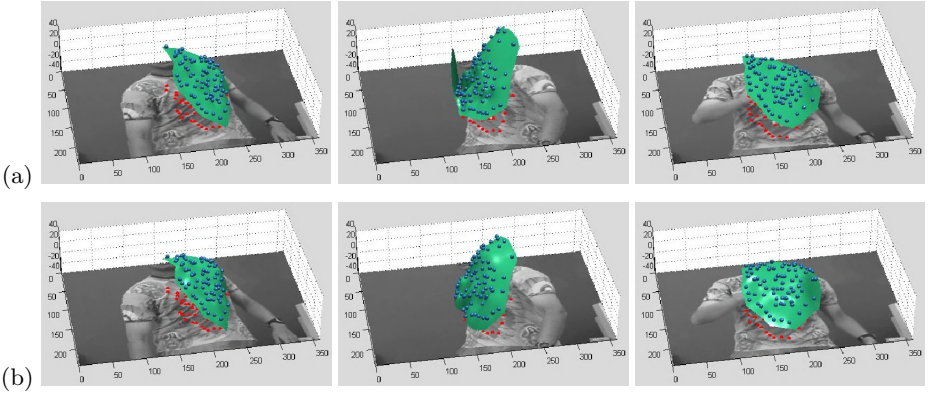



**Fig. 4.** 3D reconstruction comparison. (a) Robust covariance-weighted factorization, plus EM-Gaussian [5]. (b) Our result, using integrated non-rigid reconstruction. Note that even occluded areas are accurately reconstructed by the integrated solution.

## 5 Discussion and Future Work

We have presented techniques for tracking and reconstruction from video sequences that contain occlusions and other common violations of color constancy, as well as complicated non-rigid shape and unknown system parameters. Previously, tracking challenging footage with severe occlusions or non-rigid defor-

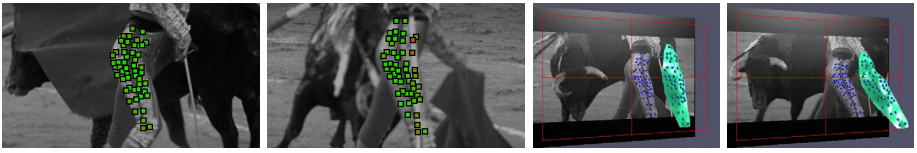


**Fig. 5.** Tracking and 3D reconstruction from a bullfight sequence, taken from the movie *Talk To Her*. (The camera is out-of-focus in the second image).

mations could only be achieved with very strong shape and appearance models. We have shown how to track such difficult sequences without prior knowledge of appearance and dynamics.

We expect that these techniques can provide a bridge to very practical tracking and reconstruction algorithms, by allowing one to model important variations in detail without having to model all other sources of non-constancy. There are a wide variety of possible extensions to this work, including: more sophisticated lighting models (e.g. [23]), layer-based decomposition (e.g. [6]), and temporal smoothness in motion and shape (e.g. [24,5]).

It would be straightforward to handle true perspective projection for rigid scenes in our framework, by performing bundle adjustment in the generalized M-step. Our model could also be learned incrementally in a real-time setting [16], although it would be necessary to bootstrap with a suitable initialization.

**Acknowledgements.** This work arose from discussions with Chris Bregler. Thanks to Hrishikesh Deshpande for help with data capture, and to Kyros Kutulakos for discussion. Portions of this work were performed while LT was visiting New York University, and AH was at University of Washington. LT was supported by ONR grant N00014-01-1-0890 under the MURI program. AH was supported in part by UW Animation Research Labs, NSF grant IIS-0113007, the Connaught Fund, and an NSERC Discovery Grant.

## References

1. Irani, M.: Multi-Frame Correspondence Estimation Using Subspace Constraints. Int. J. of Comp. Vision **48** (2002) 173–194
2. Torresani, L., Yang, D., Alexander, G., Bregler, C.: Tracking and Modeling Non-Rigid Objects with Rank Constraints. In: Proc. CVPR. (2001)
3. Brand, M.: Morphable 3D models from video. In: Proc. CVPR. (2001)
4. Soatto, S., Yezzi, A.J.: DEFORMOTION: Deforming Motion, Shape Averages, and the Joint Registration and Segmentation of Images. In: Proc. ECCV. Volume 3. (2002) 32–47
5. Torresani, L., Hertzmann, A., Bregler, C.: Learning Non-Rigid 3D Shape from 2D Motion. In: Proc. NIPS 16. (2003) To appear.
6. Jojic, N., Frey, B.: Learning Flexible Sprites in Video Layers. In: Proc. CVPR. (2001)
7. Horn, B.K.P.: Robot Vision. McGraw-Hill, New York, NY (1986)

8. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. 7th IJCAI. (1981)
9. Irani, M., Anandan, P.: About Direct Methods. In: Vision Algorithms '99. (2000) 267–277 LNCS 1883.
10. Bregler, C., Hertzmann, A., Biermann, H.: Recovering Non-Rigid 3D Shape from Image Streams. In: Proc. CVPR. (2000)
11. Torresani, L., Bregler, C.: Space-Time Tracking. In: Proc. ECCV. Volume 1. (2002) 801–812
12. Forsyth, D.A., Ponce, J.: Computer Vision: A Modern Approach. Prentice Hall (2003)
13. Dellaert, F., Seitz, S.M., Thorpe, C.E., Thrun, S.: EM, MCMC, and Chain Flipping for Structure from Motion with Unknown Correspondence. Machine Learning **50** (2003) 45–71
14. Black, M.J., Anandan, P.: The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer Vision and Image Understanding **63** (1996) 75–104
15. Jepson, A., Black, M.J.: Mixture models for optical flow computation. In: Proc. CVPR. (1993) 760–761
16. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust Online Appearance Models for Visual Tracking. IEEE Trans. PAMI **25** (2003) 1296–1311
17. Wang, J.Y.A., Adelson, E.H.: Representing moving images with layers. IEEE Trans. Image Processing **3** (1994) 625–638
18. Weiss, Y., Adelson, E.H.: Perceptually organized EM: A framework for motion segmentation that combines information about form and motion. Technical Report TR 315, MIT Media Lab Perceptual Computing Section (1994)
19. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. In Jordan, M.I., ed.: Learning in Graphical Models. Kluwer Academic Publishers (1998)
20. Morris, D.D., Kanade, T.: A Unified Factorization Algorithm for Points, Line Segments and Planes with Uncertainty Models. In: Proc. ICCV. (1998) 696–702
21. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: A factorization method. Int. J. of Computer Vision **9** (1992) 137–154
22. Shi, J., Tomasi, C.: Good Features to Track. In: Proc. CVPR. (1994) 593–600
23. Zhang, L., Curless, B., Hertzmann, A., Seitz, S.M.: Shape and Motion under Varying Illumination: Unifying Structure from Motion, Photometric Stereo, and Multi-view Stereo. In: Proc. ICCV. (2003) 618–625
24. Gruber, A., Weiss, Y.: Factorization with Uncertainty and Missing Data: Exploiting Temporal Coherence. In: Proc. NIPS 16. (2003) To appear.

## A Non-rigid Reconstruction Algorithm

The non-rigid reconstruction algorithm of Section 4 alternates between optimizing the VFE with respect to each of the unknowns. The linearization in Equation 14 is used to make these updates closed-form. This linearization also means that the distribution $q(\mathbf{z}_t)$ is Gaussian. We represent it with the variables $\mu_t \equiv E_{q(\mathbf{z}_t)}[\mathbf{z}_t]$ and $\phi_t \equiv E_{q(\mathbf{z}_t)}[\mathbf{z}_t\mathbf{z}_t^T]$.

We additionally define $\tilde{\mathbf{H}} = [\mathrm{vec}(\bar{\mathbf{S}}), \mathrm{vec}(\mathbf{V}_1), ..., \mathrm{vec}(\mathbf{V}_K)]$ and $\tilde{\mathbf{z}}_t = [1, \mathbf{z}_t^T]^T$; hence, $\mathbf{S}_t = \tilde{\mathbf{H}}\tilde{\mathbf{z}}_t$. Additionally, we define $\tilde{\mu}_t = E[\tilde{\mathbf{z}}_t]$ and $\tilde{\phi} = E[\tilde{\mathbf{z}}_t\tilde{\mathbf{z}}_t^T]$. $\tilde{\mathbf{H}}_j$ refers to the rows of $\tilde{\mathbf{H}}_j$ corresponding to the $j$-th scene point (i.e. $\mathbf{s}_{j,t} = \tilde{\mathbf{H}}_j\tilde{\mathbf{z}}_t$).

### A.1 Outlier Variables

We first note the following identity, which gives the expected reconstruction error for a pixel, taken with respect to $q(\mathbf{z}_t)$:

$$E_{q(\mathbf{z}_t)}[(I_w(\mathbf{p}_{t,j}) - \bar{I}_{w,j})^2] = \nabla I_w^T(\mathbf{R}_t\tilde{\mathbf{H}}_j\tilde{\phi}_t\tilde{\mathbf{H}}_j^T\mathbf{R}_t^T + 2\mathbf{R}_t\tilde{\mathbf{H}}_j\tilde{\mu}_t\mathbf{t}_t^T + \mathbf{t}_t\mathbf{t}_t^T)\nabla I_w - 2(\nabla I_w^T\mathbf{p}_{j,t}^0 + \bar{I}_{w,j} - I_w(\mathbf{p}_{j,t}^0))\nabla I_w^T(\mathbf{R}_t\tilde{\mathbf{H}}_j\tilde{\mu}_t + \mathbf{t}_t) (\nabla I_w^T\mathbf{p}_{j,t}^0 + \bar{I}_{w,j} - I_w(\mathbf{p}_{j,t}^0))^2 \quad (23)$$

We can then use this identity to evaluate the update steps for the outlier probabilities $\gamma_{w,j,t}$ and the noise variance $\sigma^2$ according to Equation 13 and 19, respectively.

### A.2 Shape Parameter Updates

The following shape updates are very similar to our previous algorithm [5], but with a specified covariance matrix for each track. We combine the virtual tracks for each frame into a single vector $\mathbf{f}_t = [\hat{\mathbf{p}}_{1,t}^T, ..., \hat{\mathbf{p}}_{J,t}^T]^T$; this vector has covariance $\mathbf{E}_t^{-1}$, which is a block-diagonal matrix containing $\mathbf{e}_{j,t}^{-1}$ along the diagonal. We also define $\bar{\mathbf{f}}_t = \text{vec}(\mathbf{R}_t\bar{\mathbf{S}})$, and stack the $J$ copies of the 2D translation as $\mathbf{T}_t = [\mathbf{t}_t^T, \mathbf{t}_t^T, ...\mathbf{t}_t^T]^T$.

Shape may be thus updated with respect to the virtual tracks as:

$$\mathbf{M}_t \leftarrow [\text{vec}(\mathbf{R}_t\mathbf{V}_1), ..., \text{vec}(\mathbf{R}_t\mathbf{V}_K)] \quad (24)$$

$$\beta \leftarrow \mathbf{M}_t^T(\mathbf{M}_t\mathbf{M}_t^T + \mathbf{E}_t^{-1})^{-1} \quad (25)$$

$$\mu_t \leftarrow \beta(\mathbf{f}_t - \bar{\mathbf{f}}_t - \mathbf{T}_t), \qquad \tilde{\mu}_t \leftarrow [1, \mu_t^T]^T \quad (26)$$

$$\phi_t \leftarrow \mathbf{I} - \beta\mathbf{M}_t + \mu_t\mu_t^T, \qquad \tilde{\phi} \leftarrow \begin{bmatrix} 1 & \mu_t^T \\ \mu_t & \phi_t \end{bmatrix} \quad (27)$$

$$\text{vec}(\tilde{\mathbf{H}}) \leftarrow \left(\sum_t(\tilde{\phi}_t \otimes ((\mathbf{I} \otimes \mathbf{R}_t^T)\mathbf{E}_t(\mathbf{I} \otimes \mathbf{R}_t)))\right)^{-1} \text{vec}\left(\sum_t(\mathbf{I} \otimes \mathbf{R}_t)^T\mathbf{E}_t(\mathbf{f}_t - \mathbf{T}_t)\tilde{\mu}_t^T\right) \quad (28)$$

$$\mathbf{t}_t \leftarrow \left(\sum_j \mathbf{e}_{t,j}\right)^{-1}\sum_j \mathbf{e}_{t,j}(\mathbf{f}_{tj} - \mathbf{R}_t(\bar{\mathbf{s}}_j + \sum_k \mathbf{V}_{kj}\mu_{tk})) \quad (29)$$

$$\mathbf{R}_t \leftarrow \arg\min_{R_t} || \sum_j((\tilde{\mathbf{H}}_j\tilde{\phi}_t\tilde{\mathbf{H}}_j^T) \otimes \mathbf{e}_{t,j})\text{vec}(\mathbf{R}_t) - \text{vec}(\sum_j(\mathbf{e}_{t,j}(\mathbf{f}_{tj} - \mathbf{t}_t)\tilde{\mu}_t^T\tilde{\mathbf{H}}_j^T))|| \quad (30)$$

where the symbol $\otimes$ denotes Kronecker product. Note that Equation 28 updates the shape basis $\bar{\mathbf{S}}$ and $\mathbf{V}$; conjugate gradient is used for this update. The rotation matrix $\mathbf{R}_t$ is updated by linearizing the objective in Equation 30 with exponential maps, and solving for an improved estimate.

# From a 2D Shape to a String Structure Using the Symmetry Set*

Arjan Kuijper[1], Ole Fogh Olsen[1], Peter Giblin[2], Philip Bille[3], and Mads Nielsen[1]

[1] Image Group, IT-University of Copenhagen, Glentevej 67, DK-2400 Copenhagen, Denmark
[2] Department of Mathematical Sciences, The University of Liverpool, Peach Street, Liverpool L69 7ZL, United Kingdom
[3] Algorithm Group, IT-University of Copenhagen, Glentevej 67, DK-2400 Copenhagen, Denmark

**Abstract.** Many attempts have been made to represent families of 2D shapes in a simpler way. These approaches lead to so-called structures as the Symmetry Set ($\mathcal{SS}$) and a subset of it, the Medial Axis ($\mathcal{MA}$). In this paper a novel method to represent the $\mathcal{SS}$ as a string is presented. This structure is related to so-called arc-annotated sequences, and allows faster and simpler query algorithms for comparison and database applications than graph structures, used to represent the $\mathcal{MA}$. Example shapes are shown and their data structures derived. They show the stability and robustness of the $\mathcal{SS}$ and its string representation.

## 1 Introduction

In 2D shape analysis the simplification of shapes into a skeleton-like structure is widely investigated. The Medial Axis ($\mathcal{MA}$) skeleton presented by Blum [1] is commonly used, since it is an intuitive representation that nowadays can be calculated in a fast and robust way. In so-called Shock Graphs, an $\mathcal{MA}$ skeleton is augmented with information of the distance from the boundary at which special skeleton points occurs, as suggested by Blum. Many impressive results on simplification, reconstruction and database search are reported, see e.g. [2,3, 4,5,6,7,8].

The $\mathcal{MA}$ is a member of a larger family, the Symmetry Set ($\mathcal{SS}$) [9], exhibiting nice mathematical properties, but more difficult to compute than the $\mathcal{MA}$. It also yields distinct branches, i.e. unconnected "skeleton" parts, which makes it hard to fit into a graph structure (like the $\mathcal{MA}$) for representation. In section 2 the definitions of these sets and related properties are given.

To overcome the complexity of the $\mathcal{SS}$ with respect to the $\mathcal{MA}$, we introduce in section 3 a sequentional data structure containing both the symmetry set and

* This work is part of the DSSCV project supported by the IST Programme of the European Union (IST-2001-35443).
`http://www.itu.dk/English/research/DoI/projects/dsscv/`

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 313–325, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the evolute of the shape, resulting in a representational structure that is less complex and more robust than the $\mathcal{MA}$-based structure, a graph. It is related to the so-called *arc-annotated sequence* [10,11], and allows faster and simpler query algorithms for comparison of objects and all kinds of object database applications.

Examples are given on a convex shape, showing the stability and robustness of the new data structure, in section 4, followed by the conclusions in section 5.

## 2 Background on Shapes

In this section we give the necessary background regarding properties of shapes, the Medial Axis, the Symmetry Set, and the labeling of points on these sets. For more details, see e.g. [9,6].

Let $\mathcal{S}(x(t), y(t))$ denote a closed 2D shape and $(.)_t = \frac{\partial(.)}{\partial t}$, then $\mathcal{N}(t) = (-y_t, x_t)/\sqrt{x_t^2 + y_t^2}$ denotes its unit normal vector, and $\kappa(t) = (x_t y_{tt} - y_t x_{tt})/\sqrt{x_t^2 + y_t^2}^3$ is its curvature. The evolute $\mathcal{E}(t)$ is given by the set $\mathcal{S} + \mathcal{N}/\kappa$. Note that as $\kappa$ can traverse through zero, the evolute moves "through" (minus) infinity. This occurs by definition only for concave shapes. An alternative representation can be given implicitly: $\mathcal{S}(x, y) = \{(x, y)|L(x, y) = 0\}$ for some function $L(x, y)$. Then the following formulae can be derived for $\mathcal{N}(x, y)$ and $\kappa(x, y)$: $\mathcal{N}(x, y) = (L_x, L_y)/\sqrt{L_x^2 + L_y^2}$ and $\kappa(x, y) = -(L_x^2 L_{yy} - 2L_x L_y L_{xy} + L_y^2 L_{xx})/\sqrt{L_x^2 + L_y^2}^3$ Although the curve is smooth and differentiable, the evolute contains non-smooth and non-differentiable *cusp* points, viz. those where the curvature is zero or takes a local extremum, respectively.

### 2.1 Medial Axis and Symmetry Set

The Medial Axis ($\mathcal{MA}$) is defined as the closure of the set of centers of circles that are tangent to the shape at at least two points and that contain no other tangent circles: they are so-called maximal circles. The Symmetry Set $\mathcal{SS}$ is defined as the closure of the set of centers of circles that are tangent to the shape at at least two points [9,12,13,14]. Obviously, the $\mathcal{MA}$ is a subset of the $\mathcal{SS}$ [14].

This is illustrated in Figure 1a. The two points $p_1$ and $p_2$ lie on a maximal circle and give rise to a $\mathcal{MA}$ and $\mathcal{SS}$ point. The points $p_1$ and $p_4$ give rise to a $\mathcal{SS}$ point.

To calculate these sets, the following procedure can be used, see Figure 1b: Let a circle with unknown location be tangent to the shape at two points. Then its center can be found by using the normalvectors at these points: it is located at the position of each point minus the radius of the circle times the normal vector at each point.

To find these two points, the location of the center and the radius, do the following: Given two vectors $p_i$ and $p_j$ (Figure 1b, with $i = 1$ and $j = 2$) pointing at two locations at the shape, construct the difference vector $p_i - p_j$.

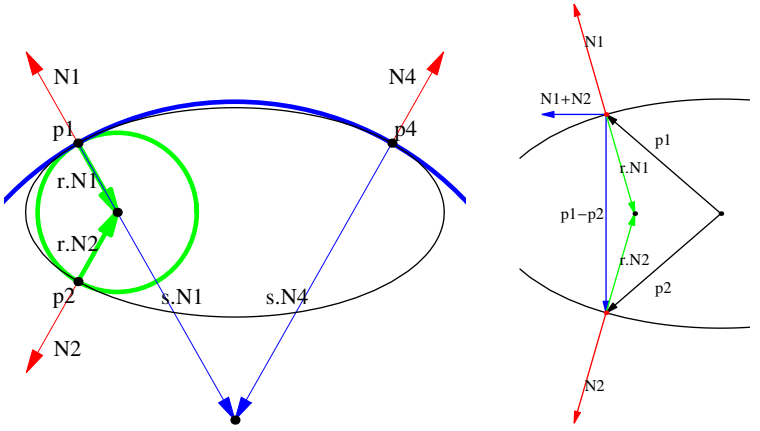



**Fig. 1.** a) Point $p_1$ contributes to two tangent circles and thus two $\mathcal{SS}$ points. Only the inner circle contributes to the $\mathcal{MA}$. b) Deriving the Medial Axis and Symmetry Set geometrically. See text for details.

Given the two unit normal vectors $N_i$ and $N_j$ at these locations, construct the vector $N_i + N_j$. If the two constructed vectors are non-zero and perpendicular,

$$(p_i - p_j).(N_i \pm N_j) = 0, \tag{1}$$

the two locations give rise to a tangent circle. The radius $r$ and the center of the circle are given by

$$p_i - rN_i = p_j \pm rN_j \tag{2}$$

and for the $\mathcal{MA}$ one only has to make sure that the circle is maximal. In the remainder of this paper we focus on the $\mathcal{SS}$.

### 2.2 Classification of Points on the Symmetry Set

It has been shown by Bruce et al. [12] that only five distict types of points can occur for the $\mathcal{SS}$, and by Giblin et al. [13,14] that they are inherited by the $\mathcal{MA}$.

- An $A_1^2$ point is the "common" midpoint of a circle tangent at two distinct points of the shape.
- An $A_1A_2$ point is the midpoint of a circle tangent at two distinct points of the shape but located at the evolute.
- An $A_1^2A_1^2$ point is the midpoint of two circles tangent at two pairs of distinct points of the shape with different radii.
- An $A_1^3$ point is the midpoint of one circle tangent at three distinct points of the shape.
- An $A_3$ point is the midpoint of a circle located at the evolute and tangent at the point of the shape with the local extremal curvature.

### 2.3 Properties of the Symmetry Set

Since the $\mathcal{SS}$ is defined locally, global properties of it are not widely investigated and difficult to derive. Banchoff and Giblin [15] have proven an invariant to hold for the number of $A_3$, $A_1A_2$, and $A_1^3$ points, both for the continuous case as the piecewise one. These numbers hold if the shape changes in such a way that the $\mathcal{SS}$ changes significantly. At these changes, called transitions [16], a so-called non-generic event for a static $\mathcal{SS}$ occurs, for instance the presence of a circle tangent to four points of the shape. Sometimes the number of $A_3$, $A_1A_2$, and $A_1^3$ points changes when the $\mathcal{SS}$ goes through a transition. For the $\mathcal{MA}$ part it implies e.g. the birth of a new branch of the skeleton. A list of possible transitions, derived from [16] is given in section 3.3.

## 3 A Linear Data Structure for the Symmetry Set

One of the main advantages of the $\mathcal{SS}$ is the possibility to represent it as a linear data structure. In general, such a structure is faster to query (according to e.g. [10,11]) than graph structures - the result of methods based on the $\mathcal{MA}$. The fact that the $\mathcal{SS}$ is a larger, more complicated set than the $\mathcal{MA}$ turns out to be advantageous in generating a simpler data structure. In this section the structure is described, together with the stability issues. Examples are given in section 4. Details on the implementations are given in [17].

### 3.1 Construction of the Data Structure

The data structure contains the elements described in the previous sections: The $\mathcal{SS}$, its special points and the evolute. They are combined in the following way (see also [17]) for an *arbitrary* planar shape:

1. Parameterize the shape.
2. Get the order of the cusps of the evolute by following the parameterization.
3. Find for the $\mathcal{SS}$ the $A_3$ points: they form the end of individual branches.
4. Relate each cusp of the evolute to an $A_3$ point.
5. Link the cusps that are on the same branch of the $\mathcal{SS}$.
6. Augment the links with labels, related to the other special points that take place when traveling from one cusp point to the other along the $\mathcal{SS}$-branch.
7. Assign the same label to different branches if an events involves the different branches: the crossings at $A_1^3$ (three identical labels) and $A_1^2/A_1^2$ (two identical labels) points. The latter can be left out, since they occur due to projection.
8. Insert moth branches (explained below) between two times two cusps as void cusps.
9. Done.

Moth branches [16] are $\mathcal{SS}$ branches without $A_3$ points. They contain four $A_1A_2$ points, that are located on the evolute. Each point is connected by the $\mathcal{SS}$ branch to two other points along the moth.

The data structure thus contains the $A_3$ points in order, links between pairs of them and augments along the links. Alternatively, one can think of a construction of a set of strings (the links), where each string contains the special points of the $\mathcal{SS}$ along the branch represented by the string. Figure 3 gives an example of the $\mathcal{SS}$ and evolute of a shape, and the derived datastructure.

### 3.2 Modified Data Structure

Since the introduction of void cusps due to moth branches violates the idea of using only the $A_3$ points as nodes, a modified structure can be used as well. In this structure the nodes contain $A_3$ and $A_1A_2$ points. These can be lined up easily, since the $A_1A_2$ points are located on the evolute between $A_3$ points. The linked connections made (strings) are now the subbranches of the $\mathcal{SS}$. The augmentation now only consists of the crossings of subbranches (either at $A_1^3$, or at both $A_1^3$ and $A_1^2/A_1^2$). This is shown in the bottom row of Figure 3.

### 3.3 Transitions

In this section the known transitions of the $\mathcal{SS}$ [16] in relation to the proposed data structure is presented. The similar thing has been done in the work on comparision of different Shock Graphs, yielding meaningful possible changes of the SG [13,4,5,6,7].

- At an $A_1^4$ transition a collision of $A_1^3$ points appears. Before and after the transition six lines, four $A_1^3$ points and three $A_1^2/A_1^2$ occur. The result on the $\mathcal{MA}$ is a reordering of the connection of two connected Y-parts of the skeleton. For the $\mathcal{SS}$, however, the Y-parts are the visible parts of $\mathcal{SS}$ branches going through $A_1^3$ points. So for the $\mathcal{SS}$ representation nothing changes.
- At an $A_1A_3$ transition, a cusp of the evolute (and thus an endpart of a $\mathcal{SS}$ branch including a $A_3$ point) intersects a branch of the $\mathcal{SS}$ and an $A_1^3$ point as well as two $A_1A_2$ points are created or annihilated. The $A_1^3$ point lies on the $A_3$ containing branch, while the other branch contains a "triangle" with the $A_1^3$ and the $A_1A_2$'s as cornerpoints: the strings $A_3[1] - a$ and $b$ change to $A_3[1] - A_1^2/A_1^2[1] - A_1^3[1] - a$ and $b_1 - A_1^3[1] - A_1A_2[1] - A_1^2/A_1^2[1] - A_1A_2[2] - A_1^3[1] - b_2$, vice versa.
- The $A_4$ transition corresponds to creation or annihilation of a swallowtail structure of the evolute and the creation or annihilation of the enclosed $\mathcal{SS}$ branch with two $A_3$ and two $A_1A_2$ points: the string $A_3[1] - A_1^2/A_1^2[1] - A_1A_2[1] - A_1A_2[2] - A_1^2/A_1^2[1] - A_3[2]$.
- At an $A_1^2A_2$ transition two non-intersecting $A_1A_2$-containing branches meet a third $\mathcal{SS}$ branch at the evolute, creating two times three different branches intersecting at two $A_1^3$ points. Or the inverse transition occurs: the strings $a$, $b_1 - A_1A_2[1] - b_2$ and $c_1 - A_1A_2[2] - c_2$ become $a_1 - A_1^3[1] - A_1^3[2] - a_3$, $b_1 - A_1^3[1] - A_1A_2[1] - A_1^2/A_1^2[1] - A_1^3[2] - b_2$ and $c_1 - A_1^3[1] - A_1^2/A_1^2[1] - A_1A_2[2] - A_1^3[2] - c_2$, vice versa.

- The $A_2^2$ moth transition describes the creation or annihilation of a $\mathcal{SS}$ branch containing only four $A_1A_2$ and no $A_3$ points. These points lie pairwise on two opposite parts of the evolute. each point is connected via the $\mathcal{SS}$ to the two points on the opposite part of the evolute: the strings $A_1A_2[1] - A_1^2/A_1^2[1] - A_1A_2[3] - A_1A_2[2] - A_1^2/A_1^2[1] - A_1A_2[4]$, if the pairs 1,2 and 3,4 are one the same part of the evolute.
- When going through an $A_2^2$ nib transition, two branches of the $\mathcal{SS}$, each containing an $A_1A_2$ point, meet and exchange a subbranch. The strings $a - A_1A_2[1] - b$ and $c - A_1A_2[2] - d$ become $a - A_1A_2[1] - c_1 - A_1^2/A_1^2[1] - c_2$ and $b_1 - A_1^2/A_1^2[1] - b_2 - A_1A_2[2] - d$.

**Stability.** The possible transitions as given above invoke only deletion, insertion or reordering of special points or branches on the data structure in an exact and pre-described manner. It is therefore a robust and stable description of the original shape.

**Arc-annotated sequences.** The structure as described above is strongly related to the so-called *arc-annotated sequences* used for RNA sequence matching and comparison. It allows the elementary edit-distance - with the insert, delete and replacement operations - as a measure of (dis)similarity between two RNA structures. The operations are directly related to the transitions as described above. For more details on this structure, the reader is referred to [10,11].

**Not for the $\mathcal{MA}$.** The string representation is *not* suitable for the $\mathcal{MA}$: Since the $\mathcal{MA}$ is a subset of the $\mathcal{SS}$, of the string only a subset of the $A_3$ points are part of the $\mathcal{MA}$ (less or equal than half of the number of points). But worse, the connections between two $A_3$ points can consist of unconnected segments. This is due to the fact that at the $\mathcal{SS}$ all local extrema of the curvature are taken into account, in contrast to the $\mathcal{MA}$.

## 4 Example: The Cubic Oval

As example shape the closed part of a cubic oval is taken, which is implicitly given by $f(x, y; a, b) = 2bxy + a^2(x - x^3) - y^2 = 0$ and $x \geq 0$. Although this is a very simple shape, it clearly shows all the possible points of the $\mathcal{SS}$ and yields a data structure that can be visually verified. Complicated shapes (e.g. those from "Shape Indexing of Image Databases (SIID)" [4]) generate strings that are too complicated to discuss in detail without having seen the elementary building blocks of the string structure.

Figure 2a shows this shape for $a = 1.025$ and $b = 0.09, 0.15, 0.30$. Changing one of these parameters, one is likely to encounter one of the transitions described above. On this shape with these values for the parameters, six extrema of the curvature occur, while the curvature doesn't change sign and the shape is thus convex. Firstly the case $a = 1.025$ and $b = 0.09$ is considered [12].

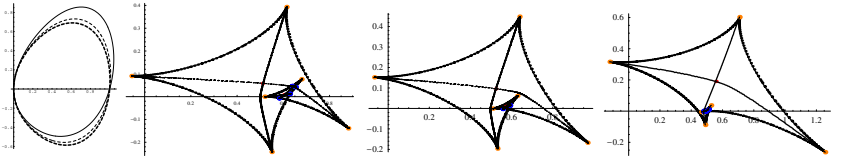


**Fig. 2.** a) The cubic oval for $a = 1.025$ and $b = .09$ (thick, dashed), $b = .15$ (intermediate thickness, dashed), and $b = .30$ (thin, continuous). The evolute and the $\mathcal{SS}$ of he cubic oval for a=1.025 and b) b=.09 c) b=.15 d) b=.30

### 4.1 Symmetry Set

The two extra extrema of the curvature, compared to the ellipse, arise from a perturbation of the shape involving an $A_4$ transition[1]. A direct consequence is that a new branch of the $\mathcal{SS}$ is created. In Figure 3b the complete $\mathcal{SS}$ with the evolute is visualized. The newly created branch has the shape of a swallowtail, as expected from the $A_4$ transition.

Since the extrema of the curvature alternate along the shape, a maximum and a minimum are created. As a prerequisite of the $A_4$ transition, the evolute is self-intersecting. Furthermore, the evolute contains six cusps. A direct consequence is that the new branch of the $\mathcal{SS}$ that is created, since branches always start in the cusps, must be essentially different from the two other branches, since the original branches start in cusps that both arise from either local maxima of the curvature, or minima. The newly created branch, however, has endpoints due to a minimum and a maximum of the curvature, so its behaviour must be different. Since real intersections - $A_1^3$ points - always involve 3 segments, a close-up is needed there.

The newly created branch introduces besides the $A_3$ and $A_1^2/A_1^2$ points the other types of special points, viz. the $A_1^3$ and $A_1A_2$ points, as shown in Figure 3a. The points are marked with dots on top of them. There are six $A_3$ points on the cusps of the evolute, four $A_1A_2$ points on the evolute close to the selfintersection part, three $A_1^2/A_1^2$ points and one $A_1^3$ point. The latter can be seen in more detail in the close-up in Figure 3b. It is close to two $A_1A_2$ points and an $A_1^2/A_1^2$ point.

### 4.2 Data Structure

To obtain the data structure, the first cusp of the evolute ($A_3$ of the $\mathcal{SS}$), is the one in the middle at the bottom. The others are taken clockwise. Then the $\mathcal{SS}$ consists of the branches $1-3$, $2-4$, and $5-6$. Branch $1-3$ intersects $2-4$ at the first $A_1^2/A_1^2$ point. The close-up of the branch $4-5$, Figure 3 toprow right, gives insight in the behaviour around this part of the $\mathcal{SS}$.

The branches $2-4$ and $5-6$ each contain two $A_1A_2$ points. Both branches intersect at an $A_1^3$ point, which is close to the two $A_1A_2$ points of branch $2-4$:

[1] For $b = 0$ and $a = .5$ an egg-shape is obtained with a $\mathcal{SS}$ similar to the ellipse, although the vertical branch is curved.

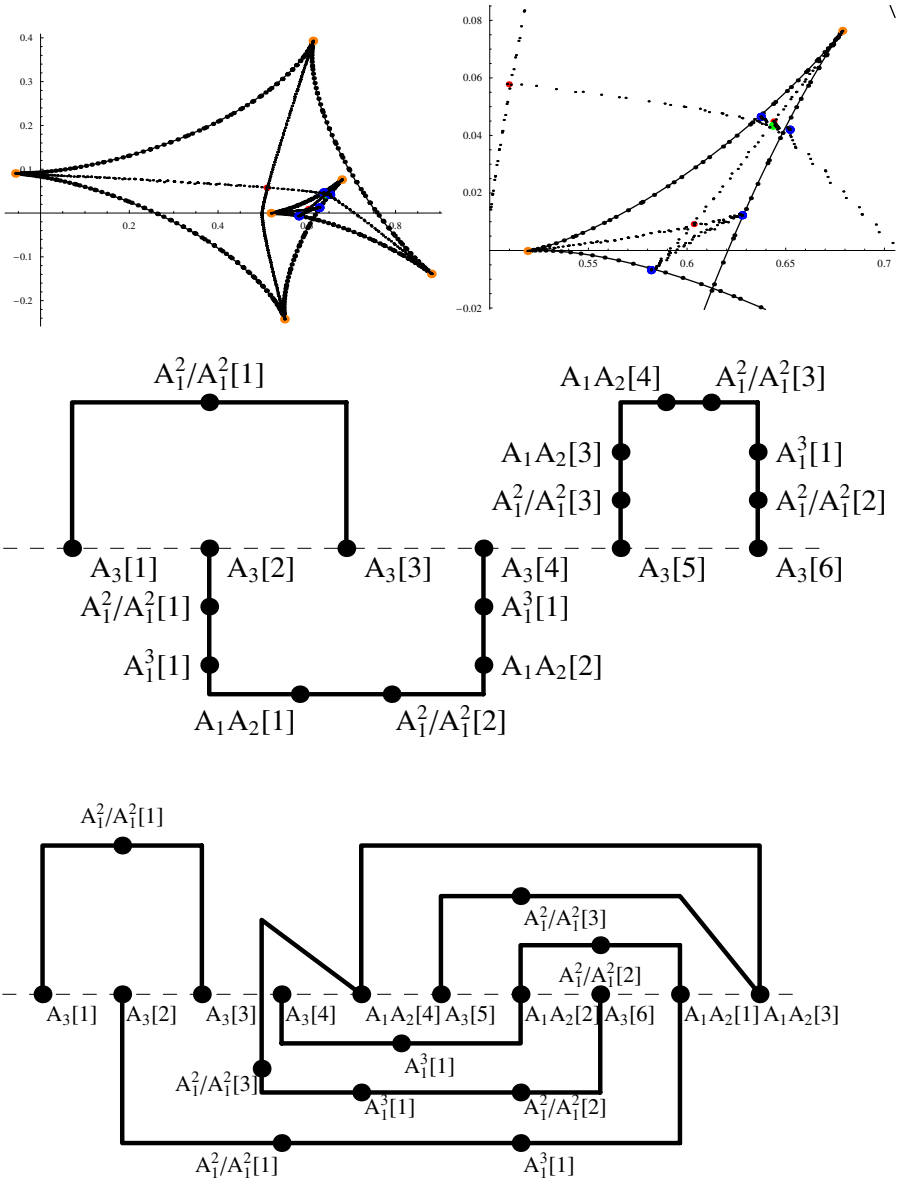



**Fig. 3.** Top: a) The evolute and the $\mathcal{SS}$ of the oval for $b = 0.09$. The contour with points is the evolute. The point $A_3[1]$, a cusp point of the evolute and an endpoint of an $\mathcal{SS}$ branch, is located at the bottom in the middle, while $A_3[3]$ is located at the top in the middle. b) Close-up, showing the $A_1^3$ point and the branch $A_3(5)$ (bottom left) - $A_3(6)$ (top right). Middle: String representation of the $\mathcal{SS}$. Bottom: Modified string representation.

At this point two subbranches of branch $2-4$ (the ones combing the $A_3$'s with the $A_1A_2$'s) and branch $4-5$ intersect. Just above this point, branch $5-6$ intersects a subbranches of branch $2-4$ (the one combing the $A_1A_2$'s) in the second $A_1^2/A_1^2$ point. Finally, two subbranches of branch $2-4$ (the ones combing the $A_3$'s with the $A_1A_2$'s) intersect at the third $A_1^2/A_1^2$ point.

So the data structure is given by the string and the links of Figure 3, middle row. The modified data structure is given by the string and the links of Figure 3, bottom row. The latter representation clearly decreases the number of augments, but increases the number of points along the string, and thus the number of links. The difference along the string between $A_3$ points and $A_1A_2$ points is due to the number of links starting and ending at a point. An $A_3$ point has one link, an $A_1A_2$ point two. Note that ignoring the projective $A_1^2/A_1^2$ points, the second data structure contains only $A_1^3$ points as augments.

The dashed horizontal line is in fact the evolute of the shape for both representations. It is cut between the points $A_3[1]$ and $A_1A_2[3]$. Therefore the representation is independent of starting point, since the two ends of the string are connected (and thus forming the evolute). It can be cut almost everywhere (albeit not at $A_3$ and $A_1A_2$ points).

### 4.3 Transitions

When $b$ is increased the shape modifies according to Figure 2a. The symmetry set changes also when $b$ is increased, as shown in Figure 2b-d. At two stages a "significant" change takes place: one of the aforementioned transitions. In the following sections the resulting symmetry sets and data structures after the transitions are given. Note that it is non-generic to encounter exactly a situation at which a transition occurs, it is only clear that transitions have been traversed.

**Annihilation of the $A_1^3$ point.** When $b$ increases to 0.15, branch $4-5$ releases branch $2-6$, see the top row of Figure 4, annihilating the involved special points. This is a typical example of an $A_1A_3$ transition. The data structures are now given by the string and the links of Figure 4, middle and bottom row.

**Creation of an $A_1^3$ point.** When $b$ increases further to $b = 0.30$, again an $A_1A_3$ transition occurs, this time the other way round. Now branch $4-5$ intersects branch $1-3$, creating the necessary involved special points, see the top row of Figure 5. The data structures are now given by the string and the links of Figure 5, middle and bottom row.

**Stability.** One can verify that the structure obtained for $b = 0.09$ and $b = 0.30$ are not identical up to rotational invariance, due to the ordering of the cusps of the evolute. With respect to the $\mathcal{MA}$ representation, the first $A_1^3$ point does not contribute to the $\mathcal{MA}$, since the $\mathcal{MA}$ consists of the connected component with the smallest radius. For $b = 0.09$, this is only a curve (the vertical oriented

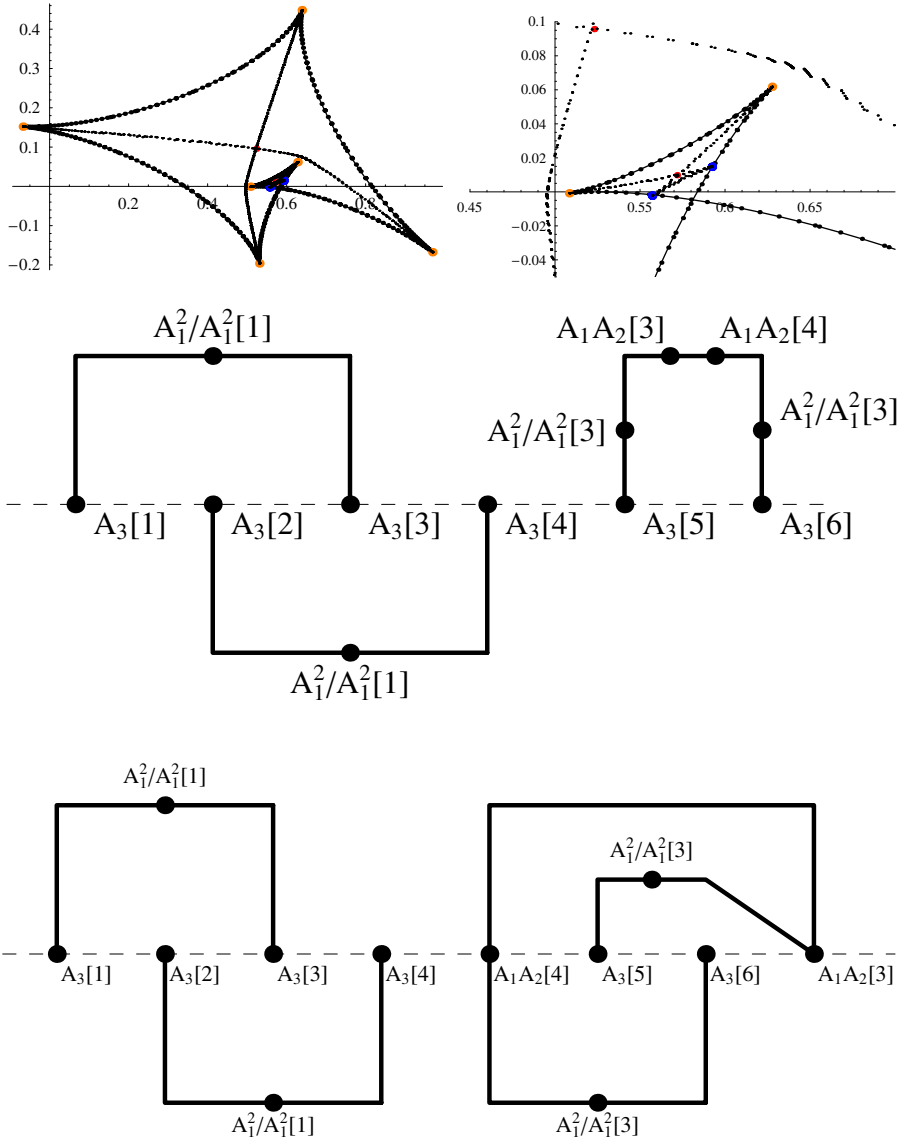



**Fig. 4.** Top: a) The evolute and the $\mathcal{SS}$ of the oval for $b = 0.15$. The contour with points is the evolute. The point $A_3[1]$, a cusp point of the evolute and an endpoint of an $\mathcal{SS}$ branch, is located at the bottom in the middle, while $A_3[3]$ is located at the top in the middle. b) Close-up, showing that the $A_1^3$ point has disappeared from the branch $A_3(5)$ (middle left) - $A_3(6)$ (top right). Middle: String representation of the $\mathcal{SS}$. Bottom: Modified string representation.

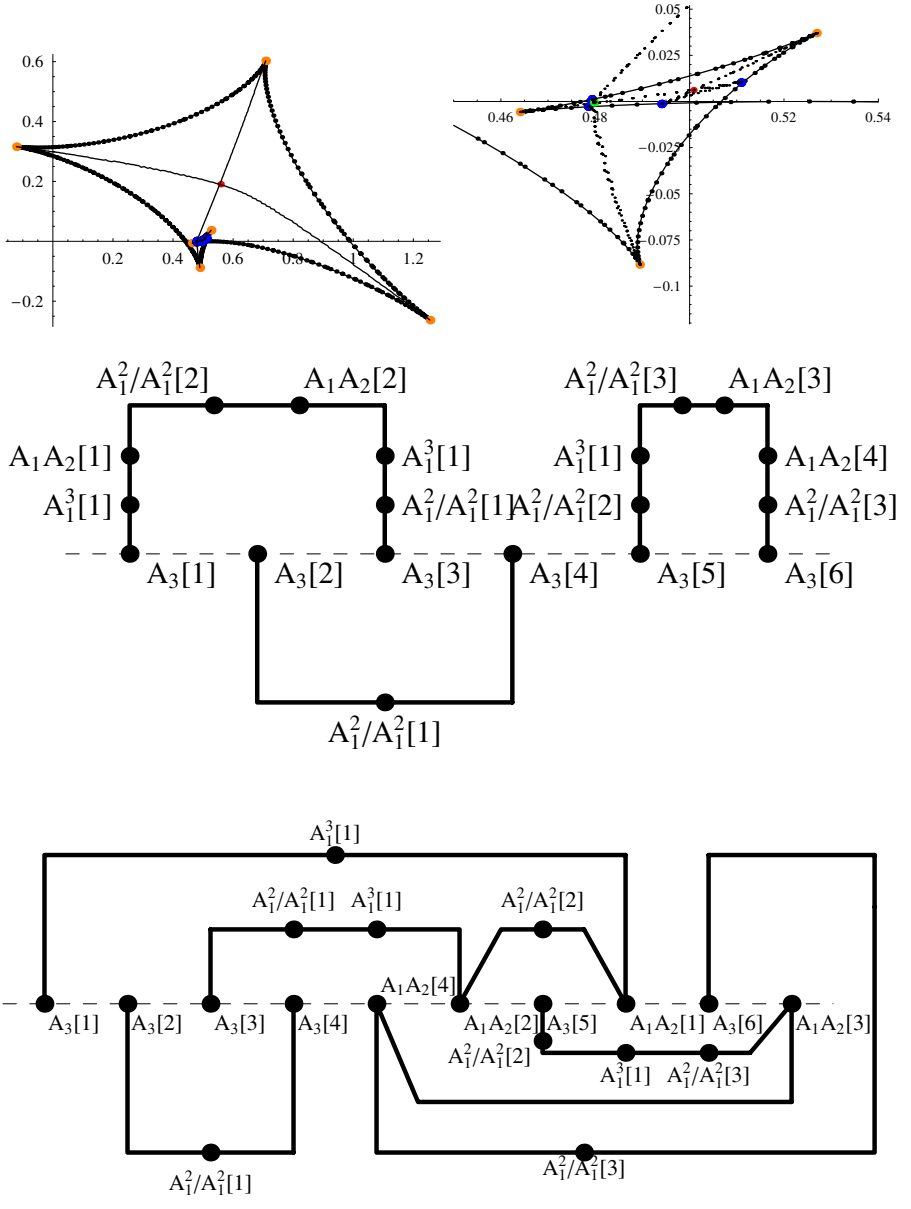



**Fig. 5.** Top:a) The evolute and the $\mathcal{SS}$ of the oval for $b = 0.30$. The contour with points is the evolute. The point $A_3[1]$, a cusp point of the evolute and an endpoint of an $\mathcal{SS}$ branch, is located at the bottom in the middle, while $A_3[3]$ is located at the top in the middle. b) Close-up, showing again an $A_1^3$ point at the branch $A_3(5)$ (middle left) - $A_3(6)$ (top right). Middle: String representation of the $\mathcal{SS}$. Bottom: Modified string representation.

one of the $\mathcal{SS}$). For $b = 0.30$, however, the swallowtail intersects this part and the $\mathcal{MA}$-skeleton now contains an extra branch, pointing from the $A_1^3$ point to the left. This event is known as an instability of the $\mathcal{MA}$, although it satisfies the known transitions. Regarding it as instability may come from the fact that the number of branches of the $\mathcal{MA}$ is not related to the number of extrema of the curvature of the shape, in contrast to the $\mathcal{SS}$. Changing the shape without changing the number of extrema of the curvature, the $\mathcal{MA}$ may gain or loose branches. The $\mathcal{MA}$ of Figure 3 and 4 is just $A_3[1] - A_3[3]$, but for Figure 5 it is formed by the 3 linesegments $A_3[1] - A_1^3[1]$, $A_1^3[1] - A_3[3]$, and $A_3[5] - A_1^3[1]$.

## 5 Conclusions

In this paper a new linear data structure representing a shape using its the symmetry set ($\mathcal{SS}$) is presented. This structure depends on the ordering of the cusps of the evolute - related to the local extrema of the curvature of the shape, as well as the $A_3$ points on the $\mathcal{SS}$. The $A_3$ points are also the endpoints of the branches of the $\mathcal{SS}$. Cusps of the evolute that are connected by the $\mathcal{SS}$ are linked in this data structure. Special points on the $\mathcal{SS}$ (where it touches the evolute, the $A_1A_2$ points, as well as intersection points - both real and those due to projection) are augmented on these links. A modified data structure takes all the points with evolute interaction - the $A_3$ and the $A_1A_2$ points - into account, again in order along the evolute.

Although the $\mathcal{SS}$ is a larger set than the Medial Axis ($\mathcal{MA}$) - even containing it - the representing *string* structure - related to the arc annotated sequence - is in complexity simpler than that for the $\mathcal{MA}$, which yields a *graph* structure. This allows (at least with respect to the theoretical complexity) faster algorithms for the comparison of different shapes, as well as (large) database queries. Obviously, a comparision study between these methods is needed, since lower computational complexity doesn't imply absolute faster query times.

The richer complexity of the $\mathcal{SS}$ prevents it from so-called instabilities that occur in the $\mathcal{MA}$. These "instabilities" are due to parts of the $\mathcal{SS}$ that "suddenly", i.e. due to certain well-known transitions for the $\mathcal{SS}$, become visible. Here the underlying $\mathcal{SS}$ influences the $\mathcal{MA}$, an argument for taking into account the complete $\mathcal{SS}$.

The only way to derive the $\mathcal{SS}$ is by means of a direct implementation of its geometric definition. The complexity of the obtained algorithm is quadratic in the number of points on the shape.

Examples of the data structures and the visualization methods are given on an example shape. Stability issues are discussed in relation to the known transitions, the significant changes of the $\mathcal{SS}$. Their description is translated in terms of the data structures, showing its stability and robustness. The proposed method is also applicable to real shapes, like outlines.

Obviously, still some theoretical questions with respect to the data structure and the $\mathcal{SS}$ are open. Although these questions are very interesting from both a theoretical and practical point of view, they don't influence the derivation and

use of the proposed data structure in it self, but may result in a speed-up of algorithms due to advanced label-checking and verification.

## References


1. Blum, H.: Biological shape and visual science (part i). Journal of Theoretical Biology **38** (1973) 205–287
2. Bouix, S., Siddiqi, K.: Divergence-based medial surfaces. In: Proceedings of the 6th European Conference on Computer Vision (2000). Volume 1842. (2000) 603–620 LNCS 1842.
3. Dimitrov, P., Phillips, C., Siddiqi, K.: Robust and efficient skeletal graphs. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000. **1** (2000) 417–423
4. Sebastian, T., Klein, P., Kimia, B.B.: Recognition of shapes by editing shock graphs. In: Proceedings of the 8th International Conference on Computer Vision (2001). (2001) 755–762
5. Siddiqi, K., Kimia, B.: A shock grammar for recognition. Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on (1996) 507–513
6. Siddiqi, K., Shokoufandeh, A., Dickinson, S., Zucker, S.: Shock graphs and shape matching. International Journal of Computer Vision **30** (1999) 1–22
7. Siddiqi, K., Bouix, S., Tannenbaum, A., Zucker, S.W.: Hamilton-jacobi skeletons. International Journal of Computer Vision **48** (2002) 215–231
8. Tek, H., Kimia, B.: Symmetry maps of free-form curve segments via wave propagation. Proceedings of the Seventh IEEE International Conference on Computer Vision, 1999. **1** (1999) 362–369
9. Bruce, J.W., Giblin, P.J.: Curves and Singularities. Cambridge University Press (1984)
10. Bafna, V., S. Muthukrishnan, S., Ravi, R.: Computing similarity between rna strings. In: Proceedings of the Sixth Symposium on Combinatorial Pattern Matching (CPM'95). (1995) 1–16 LNCS 937.
11. Jiang, T., Lin, G., Ma, B., Zhang, K.: A general edit distance between two RNA structures. Journal of Computational Biology **9** (2002) 371–388 Also appeared in RECOMB'01.
12. Bruce, J.W., Giblin, P.J., Gibson, C.: Symmetry sets. Proceedings of the Royal Society of Edinburgh **101** (1985) 163–186
13. Giblin, P.J., Kimia, B.B.: On the local form and transitions of symmetry sets, medial axes, and shocks. International Journal of Computer Vision **54** (2003) 143–156
14. Giblin, P.J., Kimia, B.B.: On the intrinsic reconstruction of shape from its symmetries. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 895–911
15. Banchoff, T., Giblin, P.J.: Global theorems for symmetry sets of smooth curves and polygons in the plane. Proceedings of the Royal Society of Edinburgh **106** (1987) 221–231
16. Bruce, J.W., Giblin, P.J.: Growth, motion and 1-parameter families of symmetry sets. Proceedings of the Royal Society of Edinburgh **104** (1986) 179–204
17. Kuijper, A.: Computing symmetry sets from 2D shapes (2003) Technical report IT University of Copenhagen no. TR-2003-36.
    `http://www.itu.dk/pub/Reports/ITU-TR-2003-36.pdf`.

# Extrinsic Camera Parameter Recovery from Multiple Image Sequences Captured by an Omni-directional Multi-camera System

Tomokazu Sato, Sei Ikeda, and Naokazu Yokoya

Graduate School of Information Science,
Nara Institute of Science and Technology,
8916-5 Takayama, Ikoma, Nara 630-0192, Japan,
{tomoka-s,sei-i,yokoya}@is.aist-nara.ac.jp,
http://yokoya.aist-nara.ac.jp/

**Abstract.** Recently, many types of omni-directional cameras have been developed and attracted much attention in a number of different fields. Especially, the multi-camera type of omni-directional camera has advantages of high-resolution and almost uniform resolution for any direction of view. In this paper, an extrinsic camera parameter recovery method for a moving omni-directional multi-camera system (OMS) is proposed. First, we discuss a perspective n-point (PnP) problem for an OMS, and then describe a practical method for estimating extrinsic camera parameters from multiple image sequences obtained by an OMS. The proposed method is based on using the shape-from-motion and the PnP techniques.

## 1 Introduction

In recent years, many types of omni-directional cameras have been developed [1, 2,3,4,5,6] and have attracted much attention in a number of different fields such as robot navigation, telepresence and video surveillance. Especially, the omni-directional multi-camera system (OMS) [4,5,6] has some advantages in the field of augmented virtuality because that provides high-resolution and almost uniform resolution for any direction of view. Generally, an OMS has multiple camera units that are located radially and are fixed in certain positions in the camera block of the OMS. Although an OMS has many advantages, the position of an OMS has been fixed in most of applications, so the camera parameter recovery of an OMS has not been well discussed in the literature. Application fields of OMS would be expanded if absolute position and posture of OMS could be recovered accurately; for example, human navigation, environment virtualization and 3-D model reconstruction. This paper provides an absolute and accurate camera parameter recovery method for a moving OMS using both a few feature landmarks and many natural features.

In a common single camera system, the extrinsic camera parameter reconstruction problem from a single image using n-point feature landmarks of known

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 326–340, 2004.
© Springer-Verlag Berlin Heidelberg 2004

3-D and 2-D positions are called the perspective n-point (PnP) problem [7,8,9]. Generally, this problem can be solved by a least-squares minimization method if six or more feature landmarks can be observed [10]. Although Chen and Chang [11] have extended this problem for generalized projection image sensors, the PnP problem for a multi-camera system has not been well discussed. Additionally, these PnP approaches cannot be successfully used if feature landmarks are partially invisible in the image sequence.

On the other hand, for a single camera system, there is another approach to recovering extrinsic camera parameters from motion of image features [12,13,14]. They are called the shape-from-motion (SFM). Although these techniques are also attempted for omni-directional camera systems [15,16,17], they cannot deal with a large number of images because such methods are sensitive to feature tracking errors and these errors must be accumulated. The problem of non-unified center of projection in an OMS has not also been well discussed.

In this paper, a practical, accurate and absolute method for extrinsic camera parameter recovery of an OMS is proposed. In our method, both the PnP and SFM techniques are used tracking both feature landmarks and natural features. The assumptions in our method are that intrinsic camera parameters (including local extrinsic parameters in camera block) are calibrated in advance and are fixed during capturing an image sequence. A small number of feature landmarks need to be visible in some frames of input for minimizing accumulative errors.

This paper is structured as follows. First, the PnP problem for an OMS at a fixed position is discussed in Section 2. Section 3 describes an extrinsic camera parameter recovery method for a moving OMS. Experimental results with real scenes and simulation will show the feasibility and accuracy of the proposed method in Section 4. Finally, Section 5 describes conclusion and future work.

## 2 PnP Problem of OMS

This section describes a method for estimating absolute extrinsic camera parameters of an OMS by solving the PnP problem. In the PnP problem, the extrinsic camera parameters are estimated by using at least six feature landmarks of known 3-D and 2-D positions. Under assumptions of this problem, intrinsic camera parameters (focal length, lens distortion parameters, center of distortion, aspect ratio) and local extrinsic parameters (relative camera positions and postures) of the camera block are known.

In the following sections, first, extrinsic camera parameters of an OMS and projection errors of feature landmarks are defined. The PnP problem is then solved to acquire the extrinsic camera parameters by dealing with multiple cameras and multiple landmarks systematically.

### 2.1 Extrinsic Camera Parameters and Projection Errors

In this section, extrinsic camera parameters of an OMS at a fixed position and projection errors of feature landmarks are defined. Generally, an OMS is composed of multiple cameras and each camera is fixed at a certain place in the OMS.

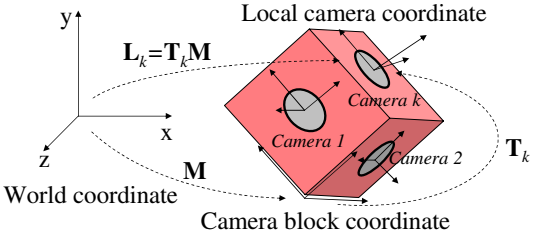



**Fig. 1.** Coordinate system of an OMS

As shown in Figure 1, the position and posture of the OMS are determined by the relationship between the camera block coordinate system and the world coordinate system, and those of each camera are determined by the relationship between the local camera coordinate system and the camera block coordinate system. In the following, the world coordinate system is transformed to the camera block coordinate system by a $4 \times 4$ matrix $\mathbf{M}$, and the camera block coordinate system is transformed to the local camera coordinate system of the $k$-th camera by the local extrinsic camera parameter $\mathbf{T}_k$.

First, 6-DOF(degree of freedom) extrinsic camera parameters (posture: $r_1, r_2$, $r_3$, position: $t_1, t_2, t_3$) from the world coordinate to the camera block coordinate are defined as follows.

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1}$$

$$= \begin{pmatrix} \mathbf{R}(r_1, r_2, r_3) & (t_1, t_2, t_3)^T \\ \mathbf{0} & 1 \end{pmatrix}, \tag{2}$$

where $\mathbf{R}$ is a $3 \times 3$ rotation matrix. The local extrinsic camera parameter $\mathbf{T}_k$ of the $k$-th camera is also defined by a $4 \times 4$ matrix in the same manner. By using $\mathbf{M}$ and $\mathbf{T}_k$, the extrinsic camera parameter $\mathbf{L}_k$ of the $k$-th camera in the world coordinate system is expressed as: $\mathbf{L}_k = \mathbf{T}_k\mathbf{M}$.

In the following expressions, for simplicity, we assume that the focal length is 1 and the lens distortion effect is already corrected. The relationship between the 2-D position $(u_p, v_p)$ on the $k$-th camera image and its 3-D position $\mathbf{S}_p = (x_p, y_p, z_p, 1)^T$ of a feature landmark $p$ in the world-coordinate system can be expressed as:

$$\begin{pmatrix} au_p \\ av_p \\ a \end{pmatrix} = \mathbf{L}_k\mathbf{S}_p = \mathbf{T}_k\mathbf{M}\mathbf{S}_p, \tag{3}$$

where, $a$ is a parameter. A computed position $(u_p, v_p)$ by Eq. (3) and an actually detected position $(\hat{u}_p, \hat{v}_p)$ of the feature landmark $p$ are not always consistent with each other due to quantization and detecting errors. The sum of squared distances between $(u_p, v_p)$ and $(\hat{u}_p, \hat{v}_p)$ for multiple $m$ landmarks has been often

used as an error function in camera parameter estimation for a single camera system [18]. In this paper, the sum of squared errors $E$ is defined as an error function for an OMS as follows.

$$E = \sum_{k=1}^{n} \sum_{p \in \mathbf{F}_k} \left\{ (u_p - \hat{u}_p)^2 + (v_p - \hat{v}_p)^2 \right\}, \tag{4}$$

where $\mathbf{F}_k$ is a set of visible landmarks from the $k$-th camera, and $n$ denotes the number of camera units in the camera block.

### 2.2 Solving the PnP Problem by Minimizing Projection Errors

This section provides a method for solving the PnP problem of the OMS, which is based on minimizing the error function $E$ defined in Eq. (4) in the previous section. By solving the PnP problem, the extrinsic camera parameter $\mathbf{M}$ of the OMS can be acquired.

The problem of computing $\mathbf{M}$ with 6-DOF by minimizing $E$ is a non-linear problem. To avoid local minimum solutions in a non-linear minimization process, a linear method is first used for computing an initial estimate of $\mathbf{M}$ without the 6-DOF constraint. After that, $\mathbf{M}$ is adjusted to 6-DOF and refined by a gradient method so as to minimize $E$ globally.

**Linear estimation of an initial parameter:** To solve the PnP problem linearly, multiple $n$ cameras and totally $j$ feature landmarks are used systematically. The local extrinsic camera parameter $\mathbf{L}_k$ of the $k$-th camera is re-defined using row vectors ($\mathbf{lx}_k$, $\mathbf{ly}_k$, $\mathbf{lz}_k$) as follows.

$$\mathbf{L}_k = \mathbf{T}_k \mathbf{M} = \begin{pmatrix} \mathbf{lx}_k \\ \mathbf{ly}_k \\ \mathbf{lz}_k \end{pmatrix}, \tag{5}$$

Eq. (3) is transformed by Eq. (5) as follows.

$$\begin{aligned} \mathbf{lx}_k \mathbf{S}_p - \hat{u}_p \mathbf{lz}_k \mathbf{S}_p = 0 \ , \\ \mathbf{ly}_k \mathbf{S}_p - \hat{v}_p \mathbf{lz}_k \mathbf{S}_p = 0 \ . \end{aligned} \tag{6}$$

Eq. (6) can be unified about $j$ points of feature landmarks, and transformed by using the parameter vector $\mathbf{m} = (m_{11}, \cdots, m_{14}, m_{21}, \cdots, m_{24}, m_{31}, \cdots, m_{34})^T$ of $\mathbf{M}$ and the local extrinsic camera parameter $\mathbf{T}_k$ as follows.

$$\mathbf{Am} = \mathbf{s}, \tag{7}$$

$$\mathbf{A} = \begin{pmatrix} s1(k_1)\mathbf{S}_1 & s2(k_1)\mathbf{S}_1 & s3(k_1)\mathbf{S}_1 \\ \vdots & \vdots & \vdots \\ s1(k_j)\mathbf{S}_j & s2(k_j)\mathbf{S}_j & s3(k_j)\mathbf{S}_j \\ s5(k_1)\mathbf{S}_1 & s6(k_1)\mathbf{S}_1 & s7(k_1)\mathbf{S}_1 \\ \vdots & \vdots & \vdots \\ s5(k_j)\mathbf{S}_j & s6(k_j)\mathbf{S}_j & s7(k_j)\mathbf{S}_j \end{pmatrix}, \mathbf{s} = \begin{pmatrix} -s4(k_1) \\ \vdots \\ -s4(k_j) \\ -s8(k_1) \\ \vdots \\ -s8(k_j) \end{pmatrix}, \tag{8}$$

$$\begin{pmatrix} s1(k_i) & s2(k_i) & s3(k_i) & s4(k_i) \\ s5(k_i) & s6(k_i) & s7(k_i) & s8(k_i) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \hat{u}_i & 0 \\ 0 & 1 & \hat{v}_i & 0 \end{pmatrix} \mathbf{T}_{k_i} \quad (i = 1, \cdots, j), \tag{9}$$

where $k_i$ is the number associated with the camera from which the feature $i$ is visible.

In Eq. (7), all the parameters except $\mathbf{m}$ are known. If $j$ is six or more, $\mathbf{m}$ can be determined linearly so as to minimize $|\mathbf{Am} - \mathbf{s}|^2$:

$$\mathbf{m} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{s}. \tag{10}$$

Note that, if distances between local cameras are much smaller than 3-D distribution of feature landmarks, computed values in Eq. (10) becomes often unstable. In this case, $\mathbf{m}'$ is defined by $m'_{ij} = m_{ij}/m_{34}$, and by approximating $\mathbf{s}$ as $\mathbf{0}$, stable values will be acquired except the scale parameter of $\mathbf{M}$ by solving $\mathbf{Am}' = \mathbf{0}$.

**Camera parameter adjustment to 6-DOF:** 12 parameters of $\mathbf{M}$ should be reduced to 3 position parameters $(t_1, t_2, t_3)$ and 3 posture parameters $(r_1, r_2, r_3)$ for Euclidean reconstruction. In this research, the position $(t_1, t_2, t_3)$ of the OMS is simply decided as $(m_{14}, m_{24}, m_{34})$. The posture parameters $(r_1, r_2, r_3)$ of the OMS can be determined from the rest of 9 parameters of rotation factors $\hat{\mathbf{R}}$ of $\mathbf{M}$ by the singular value decomposition method [19] as follows:

$$\mathbf{R}(r_1, r_2, r_3) = \mathbf{U}diag(1, 1, det(\mathbf{UV}^T))\mathbf{V}^T, \tag{11}$$

where $\mathbf{U}$ and $\mathbf{V}$ are a left and right singular vector matrices of $\hat{\mathbf{R}}$, respectively.

**Non-linear minimization of error function:** From the initial estimate of the 6 extrinsic camera parameters $(r_1, r_2, r_3, t_1, t_2, t_3)$ acquired by the previous step, The error function $E$ defined in Eq. (4) is minimized using a gradient method by iterating the following expressions until convergence.

$$r_i \leftarrow r_i - l_{r_i}\frac{\delta E}{\delta r_i}, \quad t_i \leftarrow t_i - l_{t_i}\frac{\delta E}{\delta t_i} \ (i = 1, 2, 3) \tag{12}$$

where $(l_{r_1}, l_{r_2}, l_{r_3}, l_{t_1}, l_{t_2}, l_{t_3})$ are scale factors of derivatives, and these values are decided so as to minimize $E$ in each iteration. By using this method, the extrinsic camera parameter $\mathbf{M}$ of the OMS can be determined by a few iterations so as to minimize $E$ globally with 6 DOF, because the initial estimates are expected to be close to the true parameters.

## 3 SFM of OMS

This section describes a method for estimating extrinsic camera parameters from omni-directional movies acquired by an OMS. The SFM proposed in this paper is not an ego-motion but absolute position estimation method which is based on using both feature landmarks and natural features.

The extrinsic camera parameters of the moving OMS are estimated by solving the PnP problem in each frame of input sequences using both 3-D and 2-D positions of feature landmarks and natural features. The feature landmarks are used as detonators for the 3-D position estimation of natural features. The 3-D positions of natural features for the PnP are gushed out by tracking them together. In our method, the six or more feature landmarks should be visible at least in the first and the second frames of an input video sequence to acquire the initial estimate of extrinsic camera parameters. Finally, projection errors are globally minimized for all feature landmarks and natural features.

In the following sections, first, error functions in an omni-directional movie are defined. Next the defined errors are minimized to estimate extrinsic camera parameters of the OMS in the input sequence.

### 3.1 Definition of Error Function

The sum of projection errors $E$ defined in Eq. (4) is extended for a movie input. The modified sum of projection errors in the $f$-th frame ($f = 1, \cdots, v$) of the input movie is represented by the following expression.

$$E_f = \sum_{k=1}^{n} \sum_{p \in \mathbf{F}_{kf}} W_p \left\{ (u_{fp} - \hat{u}_{fp})^2 + (v_{fp} - \hat{v}_{fp})^2 \right\}, \tag{13}$$

where $W_p$ is a confidence of feature $p$, and that is computed as an inverse covariance of projection errors of the feature $p$ [20]. $\mathbf{F}_{kf}$ is a set of feature landmarks and natural features that are visible in the $f$-th frame image of the $k$-th camera.

By using $E_f$, the total error of the input movie is defined as:

$$E_{total} = \sum_{f=1}^{v} A_f E_f, \tag{14}$$

where $A_f$ is a weighting coefficient for each frame $f$ that is set as 1 when the frame contains no specified feature landmarks or else is set to a sufficiently large value when the frame contains specified feature landmarks. In this paper, the error function $E_{total}$ is employed for estimating the extrinsic camera parameters $\mathbf{M}_f (f = 1, \cdots, v)$ and the 3-D positions $\mathbf{S}_p$ of natural features.

On the other hand, the sum of projection errors about the feature $p$ from the $fs$-th frame to the $fe$-th frame is also defined as follows:

$$EF_p(fs, fe) = \sum_{f=fs}^{fe} \left\{ (u_{fp} - \hat{u}_{fp})^2 + (v_{fp} - \hat{v}_{fp})^2 \right\}. \tag{15}$$

### 3.2 Estimation of Extrinsic Camera Parameters from an Omni-directional Movie

In this section, first, initial parameters of $\mathbf{M}_f$ and $\mathbf{S}_p$ are estimated by tracking both feature landmarks and natural features automatically by using a robust

approach. Next, $\mathbf{M}_f$ and $\mathbf{S}_p$ are then refined so as to minimize $E_{total}$ by a non-linear minimization method. The method described in this section is basically an extension of our previous work [20] for the OMS.

**Feature tracking and initial estimation:** The initial extrinsic parameter of the OMS in each frame is estimated by using the PnP technique described in the Section 2. Our method can compute the initial parameters even if the feature landmarks are invisible in most frames of a long input movie, because the substitutes of feature landmarks with known 3-D and 2-D positions are gushed out by tracking natural features. The feature landmarks are used as detonators for obtaining 3-D positions of natural features. By using a huge number of natural features to solve the PnP problem, accurate and stable estimation can be accomplished. The following paragraphs briefly describe computational steps for the $f$-th frame.

**(a) Feature tracking in each camera:** Feature landmarks are tracked automatically by a standard template matching method until a sufficient number of 3-D positions of natural features are estimated. Natural features are automatically detected and tracked by using Harris corner detector [21] for limiting feature position candidates on the images. RANSAC approach [22] is also employed for detecting outliers. In this process, these features are tracked within each camera image.

**(b) Extrinsic camera parameter estimation:** The 3-D and 2-D positions of feature landmarks and natural features are used for estimating the extrinsic camera parameter $\mathbf{M}_f$. In this step, the error function $E_f$ defined in Eq. (13) is minimized by the method described in the Section 2.2.

**(c) Feature tracking between different camera:** The features that become invisible in a certain camera are detected and tracked also in other camera images by using the extrinsic camera parameter $\mathbf{M}_f$ acquired in Step (b). The 3-D position of natural feature that has already been estimated until the previous frame is projected to each camera image by $\mathbf{L}_{kf}(= \mathbf{T}_k\mathbf{M}_f)$, and then the visibility of the 3-D position is checked. If interest points detected by Harris operator exist near by the projected position, the feature is tracked to the nearest interest point.

**(d) 3-D position estimation of natural features:** The error function $EF_p$ defined in Eq. (15) is used for estimating a 3-D position $\mathbf{S}_p$ of the feature $p$. For all the natural features tracked in the $f$-th frame, $EF_p(fs(p), f)$ is minimized and the 3-D position $\mathbf{S}_p$ is refined in every frame, where $fs(p)$ is the first detected frame of the feature $p$.

**(e) Computing confidences of natural features:** In this paper, the distribution of tracking errors of the feature is approximated by a Gaussian probability density function. Then, the confidence $W_p$ of the feature $p$ is computed as an inverse covariance of projection errors from the $fs(p)$-th frame to the $f$-th frame, and refined in every frame [20].

**(f) Addition and deletion of natural features:** In order to obtain accurate estimates of camera parameters, good features should be selected. In

this paper, the set of natural features is automatically updated by checking conditions of features using multiple measures [20].

By iterating the steps above from the first frame to the last frame of the input movie, the initial estimate of the extrinsic camera parameters of the OMS is computed.

**Global optimization in video sequence:** From the initial parameters of $\mathbf{M}_f (f = 1, \cdots, v)$, the error function $E_{total}$ defined in Eq. (14) is gradually minimized by using derivatives of parameters. This minimization process is almost the same as the method in Section 2.2, except for the 3-D positions of the natural features $\mathbf{S}_p$. In this minimization, the 3-D positions $\mathbf{S}_p = (x_p, y_p, z_p, 1)$ of natural features are also adjustable parameters and refined by using derivatives $(\frac{\delta E_{total}}{\delta x_p}, \frac{\delta E_{total}}{\delta y_p}, \frac{\delta E_{total}}{\delta z_p})$. The feature confidences $W_p$ computed in the iterating process in each frame are also used for this error function $E_{total}$.

By iterating this minimization for all the input images until convergence, an accurate extrinsic camera parameters and 3-D positions of natural features can be acquired. Local minimum and computational cost problems can be avoided simply by a standard gradient descent method, because the initial parameters are expected to be sufficiently close to the true values.

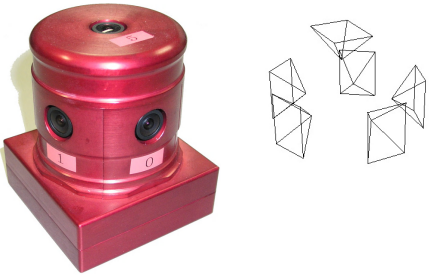
## 4 Experiments

In this section, two kinds of experimental results are demonstrated. In the first experiment, the accuracy of extrinsic parameters estimated by the method described in Section 2 is evaluated by computer simulation. The second experiment is concerned with 3-D reconstruction test in real environments.

In all the experiments, Ladybug camera system [23] is used as an OMS. As shown in Figure 2, Ladybug has radially located six camera units in the camera block and their positions and postures are fixed. Each camera can acquire $768 \times 1024$ resolution images at 15 fps, and the multi-camera system can capture a scene covering more than 75% of full spherical view. The intrinsic parameters of the camera block are estimated as shown in Figure 2(b) by using a laser measurement system and a calibration board [6] in advance. From the result of camera calibration, it is known that displacements of adjacent camera units are $40 \pm 2$mm in the horizontal direction and $46 \pm 4$mm in the vertical direction.

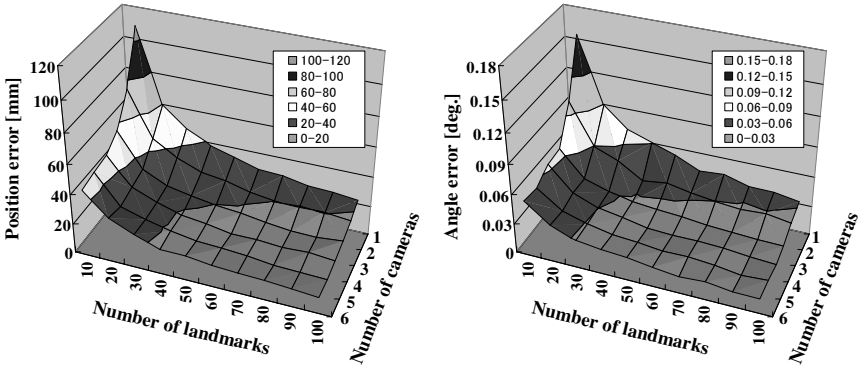
### 4.1 Quantitative Evaluation of Solving PnP Problem in Simulation

In this section, the effectiveness of the OMS for the PnP problem is quantitatively evaluated by computer simulation. This simulation is carried out by using a virtual Ladybug at a fixed position and feature landmarks that are randomly scattered in 50m to 100m range of space from the virtual Ladybug. The feature landmarks are projected to each camera of the virtual Ladybug, and are detected with a Gaussian noise. The Gaussian noise is set so as to have 2.0 pixel standard



(a) appearance (b) viewing volume

**Fig. 2.** Omni-directional camera system "Ladybug".




(a) position error (b) angle error

**Fig. 3.** Errors in camera block position and angle (simulation).

deviation in each image. After this process, projected positions are quantitized on pixels.

In this situation for solving the PnP problem, both the number of landmarks and the number of cameras of the OMS are changed, and position errors and angle errors by the method described in Section 2 are measured. Figure 3 (a) and (b) show the computed average errors in camera position and angle with a hundred trials. It should be noted that both (a) and (b) exhibit the same behavior: The average error monotonously decreases when the number of landmarks and cameras are increased. Especially about the number of cameras, it is confirmed that the use of the OMS is effective for solving the PnP problem accurately comparing with a single camera system.

### 4.2 Experiments with Real Scenes

To demonstrate the validity of the proposed method described in Section 3, extrinsic camera parameters of moving Ladybug are actually reconstructed and evaluated in an indoor and outdoor environment. For both experiments, some

of natural features are used as feature landmarks and their 3-D positions are measured by the total station (Leica TCR1105XR). These feature landmarks are specified manually on the first frame image and the last frame image.

**Camera parameter recovery for an indoor scene:** In this experiment, an indoor scene is captured as an image sequence of 450 frames for each camera by walking in a building as shown in Figure 4. First, the extrinsic camera para-

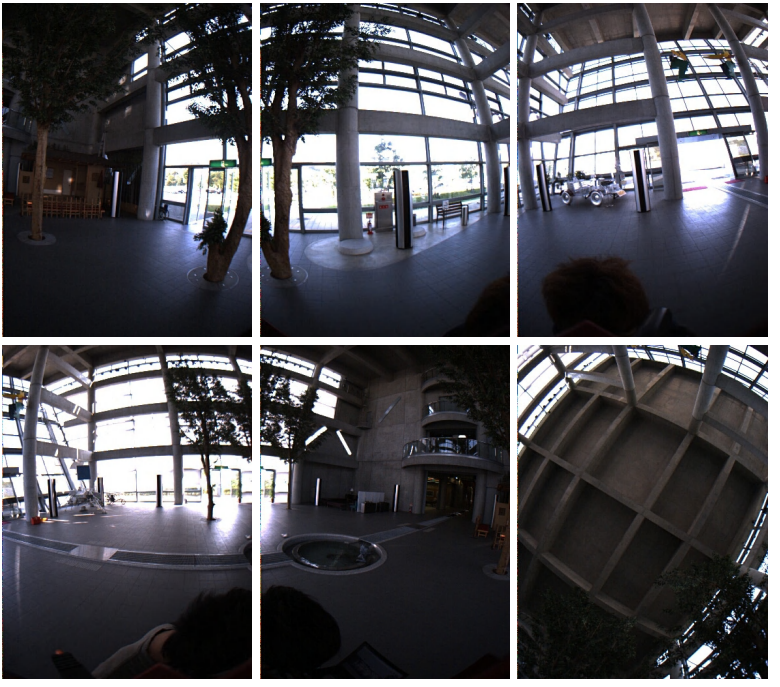
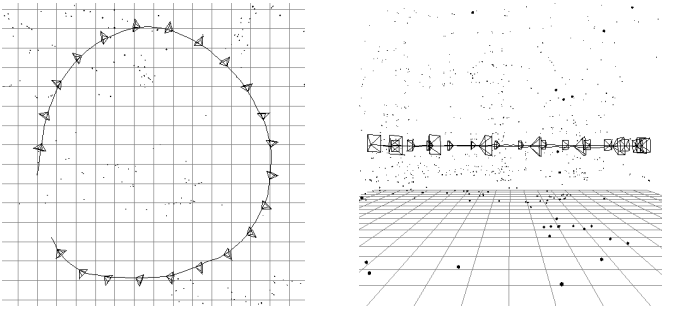


**Fig. 4.** Sampled frames of input image sequences obtained by six cameras (indoor scene).



(a) top view (b) side view

**Fig. 5.** Result of extrinsic camera parameter estimation (indoor scene).

meters of Ladybug are reconstructed by the proposed method. On an average, approximately 440 points of natural features are automatically tracked in each set of frames by six cameras. The squared average of re-projection errors of the features is 2.1 pixels. Figure 5 shows the recovered extrinsic camera parameters of the camera 1 of the Ladybug. The curved line in this figure indicates the camera path, the quadrilateral pyramids indicate the camera postures drawn at every 20 frames. The black point clouds show the estimated 3-D positions of the natural features. The length of the recovered camera path is 29m. As shown in this figure, the camera parameters are recovered very smoothly.

**Camera parameter recovery for an outdoor scene:** An outdoor scene is captured by walking in our campus including several buildings as shown in Figure 6. The image sequence for each camera consists of 500 frames. The distances between the camera system and objects in this scene are longer than those in the indoor scene. In this experiment, approximately 530 points of natural features on an average are automatically tracked in each set of frames by six cameras. The squared average of re-projection errors of the features is 1.6 pixels. Figure 7 shows the recovered extrinsic camera parameters of the camera 1 of the Ladybug. The length of the recovered camera path is also 29m. As same as the indoor environment, the camera parameters are recovered very smoothly.

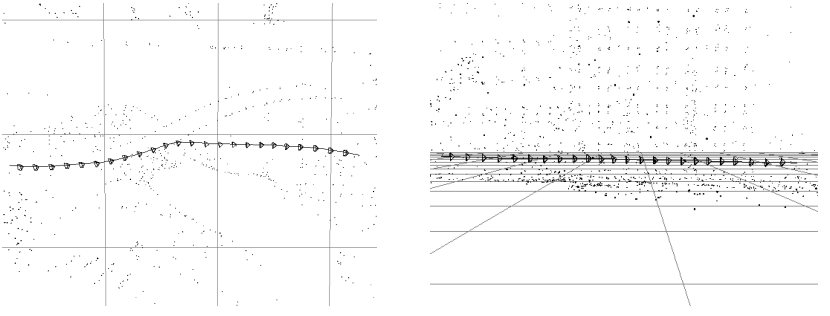
**Quantitative evaluation for real scenes:** The recovered camera paths and postures are evaluated by comparing with the ground truth. The ground truth is made by solving the PnP problem in every frame. For obtaining the ground truth, features in the input images are manually tracked throughout the input sequences and their 3-D positions are measured by the total station.

Figure 8 denotes position errors and posture errors for the indoor data. The average estimation errors in position and posture of the camera system before global optimization are 50mm and 0.49degree, respectively. After global optimization, they are reduced to 40mm and 0.12degree, respectively. We can confirm that the accumulation of estimation errors is reduced by global optimization. As same as for the indoor sequence, Figure 9 illustrate position errors and posture errors about the outdoor scene. In this sequence, the average estimation error before global optimization is 280mm(position) and 1.10degrees(angle). After the global optimization, they are reduced to 170mm(position) and 0.23degrees(angle), respectively. It is also confirmed that the accumulation of the estimation errors is effectively reduced by global optimization. Note that the average errors of the outdoor scene are larger than those of the indoor scene, because the scale of the outdoor scene is several times larger than the scale of the indoor scene. Although these errors are considered as significantly small comparing with the scale of each scene, more accurate reconstruction could be accomplished by specifying additional feature landmarks, for example, in the middle frame.

## 5 Conclusion

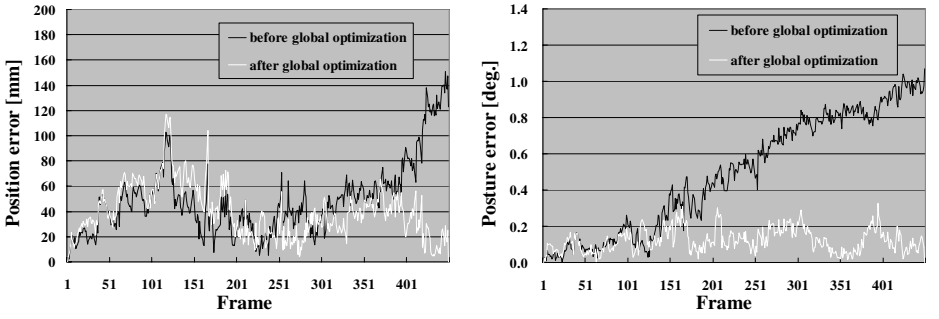
This paper has proposed a method for recovering extrinsic camera parameters of an OMS. In the proposed method, first, the PnP problem for an OMS is solved to recover an extrinsic camera parameter using feature landmarks. Next, extrinsic parameters of the OMS are estimated by using an SFM approach which is based on tracking both feature landmarks and natural features.

**Fig. 6.** Sampled frames of input image sequences obtained by six cameras (outdoor scene).



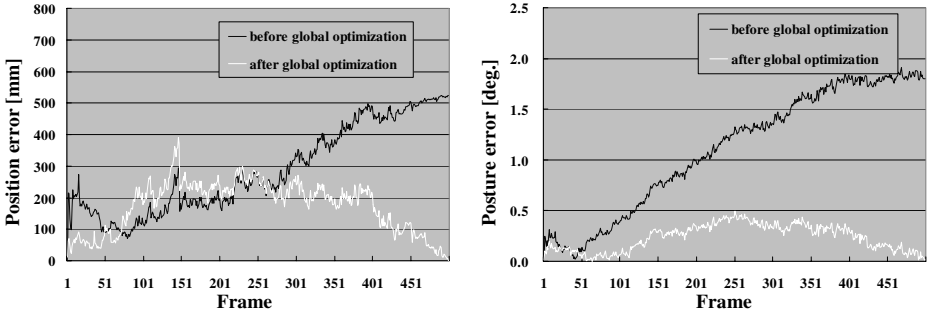
(a) top view (b) side view

**Fig. 7.** Result of extrinsic camera parameter estimation (outdoor scene).




(a) position error (b) posture error

**Fig. 8.** Errors in estimated camera path and posture (indoor scene).




(a) position error (b) posture error

**Fig. 9.** Errors in estimated camera path and posture (outdoor scene).

In the experiments, the effectiveness of the use of an OMS for PnP problem is quantitatively examined by computer simulation. Additionally, the extrinsic camera parameter recovery with real scenes is successfully demonstrated using multiple long image sequences captured by a real OMS: Ladybug. In future work, the recovered camera parameters will be applied for dense 3-D scene reconstruction of outdoor environments.

## References

1. K. Miyamoto: "Fish Eye Lens," Jour. of Optical Society of America, Vol. 54, No. 2, pp. 1060–1061, 1964.
2. K. Yamazawa, Y. Yagi and M. Yachida: "Omnidirectional Imaging with Hyperboloidal Projection," Proc. Int. Conf. on Intelligent Robots and Systems, Vol. 2, pp. 1029–1034, 1993.
3. S. K. Nayar: "Catadioptic Omnidirectional Cameras," Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition, pp. 482–488, 1997.

4. J. Shimamura, H. Takemura, N. Yokoya and K. Yamazawa: "Construction and Presentation of a Virtual Environment Using Panoramic Stereo Images of a Real Scene and Computer Graphics Models," Proc. 15th IAPR Int. Conf. on Pattern Recognition, Vol. IV, pp. 463–467, 2000.
5. H. Tanahashi, K. Yamamoto, C. Wang and Y. Niwa: "Development of a Stereo Omni-directional Imaging System(SOS)," Proc. IEEE Int. Conf. on Industrial Electronics, Control and Instrumentation, pp. 289–294, 2000.
6. S. Ikeda, T. Sato and N. Yokoya: "High-resolution Panoramic Movie Generation from Video Streams Acquired by an Omnidirectional Multi-camera System," Proc. IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent System, pp. 155–160, 2003.
7. R. Horand, B. Conio and O. Leboullex: "An Analytic Solution for the Perspective 4-Point Problem," Computer Vision, Graphics, and Image Processing, Vol. 47, pp. 33–44, 1989.
8. J. S. C. Yuan: "A General Photogrammetric Method for Determining Object Position and Orientation," IEEE Trans. on Robotics and Automation, Vol. 5, No. 2, pp. 129–142, 1989.
9. R. Krishnan and H. J. Sommer: "Monocular Pose of a Rigid Body Using Point Landmarks," Computer Vision and Image Understanding, Vol. 55, pp. 307–316, 1992.
10. R. Klette, K. Schluns and A. koschan Eds.: Computer Vision: Three-dimensional Data from Image, Springer, 1998.
11. C. S. Chen and W. Y. Chang: "Pose Estimation for Generalized Imaging Device via Solving Non-perspective N Point Problem," Proc. IEEE Int. Conf. on Robotics and Automation, pp. 2931–2937, 2002.
12. P. Beardsley, A. Zisserman and D. Murray: "Sequential Updating of Projective and Affine Structure from Motion," Int. Jour. of Computer Vision, Vol. 23, No. 3, pp. 235–259, 1997.
13. C. Tomasi and T. Kanade: "Shape and Motion from Image Streams under Orthography: A Factorization Method," Int. Jour. of Computer Vision, Vol. 9, No. 2, pp. 137–154, 1992.
14. M. Pollefeys, R. Koch, M. Vergauwen, A. A. Deknuydt and L. J. V. Gool: "Three-dimentional Scene Reconstruction from Images," Proc. SPIE, Vol. 3958, pp. 215–226, 2000.
15. J. Gluckman and S. Nayer: "Ego-motion and Omnidirectional Cameras," Proc. 6th Int. Conf. on Computer Vision, pp. 999–1005, 1998.
16. M. Etoh, T. Aoki and K. Hata: "Estimation of Structure and Motion Parameters for a Roaming Robot that Scans the Space," Proc. 7th Int. Conf. on Computer Vision, Vol. I, pp. 579–584, 1999.
17. C. J. Taylor: "VideoPlus," Proc. IEEE Workshop on Omnidirecitonal Vision, pp. 3–10, 2000.
18. B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon: "Bundle Adjustment a Modern Synthesis," Proc. Int. Workshop on Vision Algorithms, pp. 298–372, 1999.
19. K. Kanatani: Statistical Optimization for Geometric Computation: Theory and Practice, Elsevier Science, 1998.
20. T. Sato, M. Kanbara, N. Yokoya and H. Takemura: "Dense 3-D Reconstruction of an Outdoor Scene by Hundreds-baseline Stereo Using a Hand-held Video Camera," Int. Jour. of Computer Vision, Vol. 47, No. 1-3, pp. 119–129, 2002.
21. C. Harris and M. Stephens: "A Combined Corner and Edge Detector," Proc. Alvey Vision Conf., pp. 147–151, 1988.

22. M.A. Fischler and R.C. Bolles: "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Communications of the ACM, Vol. 24, No. 6, pp. 381–395, 1981.
23. Point Gray Research Inc.: "Ladybug," http://www.ptgrey.com/products/ladybug/index.html.

# Evaluation of Robust Fitting Based Detection

Sio-Song Ieng[1], Jean-Philippe Tarel[2], and Pierre Charbonnier[3]

[1] LIVIC (LCPC-INRETS), 14, route de la minière, Bât 824,
F-78000 Versailles-Satory, France
ieng@inrets.fr
[2] ESE (LCPC), 58,boulevard Lefèbvre,
F-75732 Paris Cedex 15 France
tarel@lcpc.fr
[3] LRPC de Strasbourg,
11, Rue Jean Mentelin, BP 9,
F-67200 Strasbourg, France
Pierre.Charbonnier@equipement.gouv.fr

**Abstract.** Low-level image processing algorithms generally provide noisy features that are far from being Gaussian. Medium-level tasks such as object detection must therefore be robust to *outliers*. This can be achieved by means of the well-known M-estimators. However, higher-level systems do not only need robust detection, but also a confidence value associated to the detection. When the detection is cast into the fitting framework, the inverse of the covariance matrix of the fit provides a valuable *confidence matrix*.
Since there is no closed-form expression of the covariance matrix in the robust case, one must resort to some approximation. Unfortunately, the experimental evaluation reported in this paper on real data shows that, among the different approximations proposed in literature that can be efficiently computed, none provides reliable results. This leads us to study the robustness of the covariance matrix of the fit with respect to noise model parameters. We introduce a *new non-asymptotic approximate covariance matrix* that experimentally outperforms the existing ones in terms of reliability.

## 1 Introduction

In modern applications, such as intelligent transportation systems, cameras and their associated detection algorithms are increasingly considered as specialized sensors. As for every sensor, the vision measure must be accompanied with a *confidence* value in order to be integrated into a higher level system. This is especially important when safety aspects are involved. But then the point is: how can this evaluation be performed?

Many image analysis algorithms, such as image segmentation or curve detection, can be at least partially formalized as *fitting* problems. Least-squares fitting, the well known technique based on the assumption of Gaussian noise, is widely used, and it provides the exact covariance matrix of the obtained fit. The

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 341–352, 2004.
© Springer-Verlag Berlin Heidelberg 2004

inverse of this matrix can be used as a *confidence matrix*. However, it is common knowledge that, *in real applications, the noise is seldom Gaussian*, making the fitting task more difficult. Alternatively, using the robust framework, it is possible to deal with heavy tailed noise models, but this leads to non-linear equations and iterative algorithms such as reweighted least-squares, in the context of *M-estimators* [1].

Although robust fitting algorithms have been extensively investigated, the problem of deriving a confidence matrix has seldomly been considered in image analysis. Indeed, due to non-linearities, an exact derivation of the covariance matrix is believed to be intractable in the robust framework and approximations are required. One way to tackle the problem is to study the asymptotic behavior of robust estimators: Huber [1] proposes several such approximate covariance matrices. The evaluation of these matrices was only performed on synthetic data. In a different context, namely robust Kalman filtering, where the question of the predicted covariance matrix becomes of major importance, other approximations were proposed in [2] and in [3], but without justification. Again, the evaluation of these matrices was performed only on synthetic data.

We started our study by an experimental comparison of the approximate covariance matrices already proposed in the literature. The results of this comparison on real and simulated data showed that none of them gives sufficiently reliable results when noise is far from being Gaussian, which led us to derive a new approximate covariance matrix. Unlike those proposed by Huber, it is not an asymptotic covariance matrix. However, its experimental performances are much more satisfactory.

Robust fitting using M-estimators is summarized in Section 2, where the choice of the noise model and its parameters is also discussed. Then in Section 3, a new approximation is derived and the experimental comparison of approximate covariance matrices is described.

## 2 Fitting Based Detection

We focus, as an application, on lane-markings detection in images [3]. For each image the detection algorithm consists of a feature extraction step followed by a robust curve fitting. The feature extractor scans each row $x$ of the image and provides a series of coordinates $(x_i, y_i)$, $i = 1, \cdots, n$ that should correspond to the center of each lane-marking. The detection can then be seen as a fitting problem, using explicit curves along one axis of the image. The assumed link between a couple of coordinates $(x_i, y_i)$ is

$$y_i = X_i^t A + b_i. \tag{1}$$

In this model, the measurement noise, $b$ is assumed to be centered, independent and identically distributed (iid). The vector $X_i = (x_i^k)$, $k = 0, \cdots, d$ is the vector of monomials. While a straight line is sufficient as a lane-marking model for near-field cameras, other basis functions could be used to model more complex

lane-marking curves, provided the relationship remains linear with respect to the curve parameters, $A = (a_k)$, $k = 0, \cdots, d$.

In the remainder of the section, the robust fitting procedure is reviewed and its performances and limitations are discussed.

### 2.1 Feature Noise Model

An experimentally convenient way of modeling observed noise distribution is to use the so-called Smooth Exponential Family $F_{\alpha,s}(b)$ of functions introduced in [3]:

$$F_{\alpha,s}(b) \propto \frac{1}{s} e^{-\frac{1}{2}\phi_\alpha(\frac{b^2}{s^2})} \tag{2}$$

where

$$\phi_\alpha(\frac{b^2}{s^2}) = \frac{1}{\alpha}((1 + \frac{b^2}{s^2})^\alpha - 1).$$

The two parameters of this family are $s$ and the power $\alpha$. The former is the scale parameter, while the latter specifies the shape of the noise distribution. Indeed, $\alpha$ allows a continuous transition between well-known statistical laws such as Gauss ($\alpha = 1$), and Geman & McClure [4] ($\alpha = -1$). Another advantage of this family is that it only contains differentiable functions allowing robust fitting, as proved in [3]. Moreover, $F_{\alpha,s}$ can always be normalized on a bounded support, so it can still be seen as a probability distribution function (pdf). In this family, when $\alpha$ decreases, the probability of observing large errors (outliers) increases.

### 2.2 Robust Fitting

Following the MLE approach and assuming the noise model (2), the problem is equivalent to minimizing, with respect to $A$, the error:

$$e(A) = \frac{1}{2} \sum_{i=1}^{i=n} \phi_\alpha((\frac{X_i^t A - y_i}{s})^2).$$

A local minimum is achieved by the classical alternate minimisation scheme:

1. Initialize $A_0$, and set $j = 1$.
2. For all indexes $i$ ($1 \leq i \leq n$), compute weights $\lambda_{i,j} = \phi'_\alpha((\frac{X_i^t A_{j-1} - y_i}{s})^2)$.
3. Solve the linear system $\sum_{i=1}^{i=n} \lambda_{i,j} X_i X_i^t A_j = \sum_{i=1}^{i=n} \lambda_{i,j} X_i y_i$,
4. If $\|A_j - A_{j-1}\| > \epsilon$, increment $j$, and go to 2, else a local minimum is achieved.

This algorithm was first introduced for M-estimators by Huber [1]. A proof of the local convergence of the previous algorithm, using Kuhn and Tucker's theorem is described in [3]. When $\alpha = 0$, it is easy to show that generalized T-Student [5] pdfs are used as noise model.

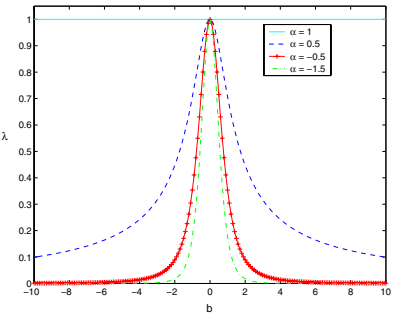



**Fig. 1.** Variations of weight $\lambda$ with respect to the value $b$ of the noise, for different values of $\alpha$. Outliers effect is strongly reduced by this weight during the fitting.

### 2.3 Influence of $\alpha$ and $s$ on Fits

The choices of the parameters of the noise model, $\alpha$ and $s$, are now discussed.

Fig. 1 shows that the weight $\lambda$ becomes less sharply peaked when $\alpha$ is decreasing. The same kind of figure is obtained with increasing $s$. As a consequence, the lower $\alpha$ or $s$ is, the lower the effect of outliers on the result is.

But one must be careful not decrease $\alpha$ or $s$ too much. Indeed, with too small an $\alpha$ or $s$, the number of local minima increases and the curve fitting algorithm has a higher chance of being trapped in a local minimum located far from the global one. In practice, this implies that the estimated curve parameters are *frozen* at their current values during the alternate minimization [6]. So, how can the scale and power parameters be correctly estimated?

When the pdf support is unbounded ($\alpha \geq 0$) and other noise parameters are known, by deriving the negative log-likelihood with respect to $s$, we get the following implicit equation in $\hat{s}$:

$$\hat{s}^2 = \frac{1}{n} \sum_{i=1}^{i=n} \phi'((\frac{X_i^t A - y_i}{\hat{s}})^2)(X_i^t A - y_i)^2. \tag{3}$$

This equation means that the MLE estimate $\hat{s}$ of $s$ is a fixed point. When $\alpha < 0$, a similar expression, accounting for bounded support pdf's can be also derived. Other scale estimators such as MAD [7] have been proposed.

The MLE estimator of the scale must be used with care. Surprisingly, when data has little noise, we observed that the MLE estimate of $s$ is clearly under-estimated. This is a general problem, due to the finite precision of feature detectors (in our case, it provides discrete image positions). In Fig. 2, we simulate this effect by generating a Cauchy noise ($\propto \frac{1}{1+(\frac{b}{s})^2}$ or $\alpha = 0$) with a fixed scale $s$, and rounding it. Then, the MLE scale $\hat{s}$ is estimated from this data, assuming a Cauchy noise with unknown scale. Fig. 2 confirms our observations on real data: when the true scale $s$ is lower that one pixel, $\hat{s}$ is clearly under-estimated.

This suggests that during its estimation, the scale must not been allowed to take small values (i.e. lower than one pixel). This also implies that it is better

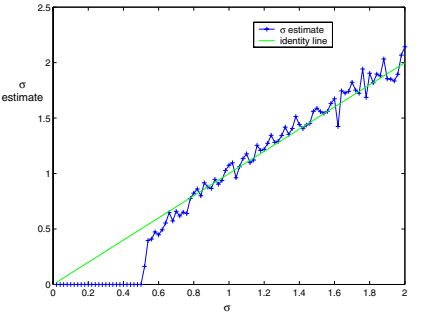



**Fig. 2.** The estimated scale $\hat{s}$ versus the true scale $s$ for Cauchy noise. We can notice that $\hat{s}$ is under-estimated by the MLE estimator when $s < 1$, due to rounding the data.

not to estimate the scale between each step of the robust fitting, contrary to what was suggested in [1]. Indeed, an under-estimated scale can freeze the fit in its current state, while it has not yet converged.

The MLE approach, when applied to the estimation of $\alpha$, assuming $s$ is known, does not seem to lead to closed-form estimators. Nevertheless, this estimation may always be performed by minimizing numerically the likelihood with respect to $\alpha$, using a minimizer such as a Gradient Descent algorithm. In our experiments, we have obtained $\hat{\alpha} = 0.05$ on the residual noise collected on 150 images after fitting.

### 2.4 Robustness with Respect to $\alpha$ and $s$

By nature, robust fitting algorithms are not very sensitive to outliers. However, another question of importance is their robustness to an inappropriate choice of the noise model parameters. In order to evaluate the effect of this kind of modeling error on the fitting, we run the following experiment:

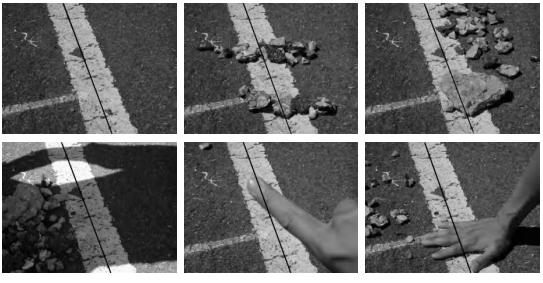


**Fig. 3.** Six images from the set of 150 images used in the experiments. The black straight line is the reference fit.

1. Collect a set of 150 images showing the same lane-marking in the same position but with different perturbation (Several images are shown in Fig. 3).

2. The reference lane-marking is accurately measured by hand. In this experiment fits are straight lines.
3. For each image of the set, fits are estimated for pairs of $(\alpha, s)$ ranging on a grid. The initial curve parameters, $A_0$, are set to random values.
4. The relative error between the fits and the reference is averaged on the image set for each pair $(\alpha, s)$. Fig. 4 shows the obtained error surface.

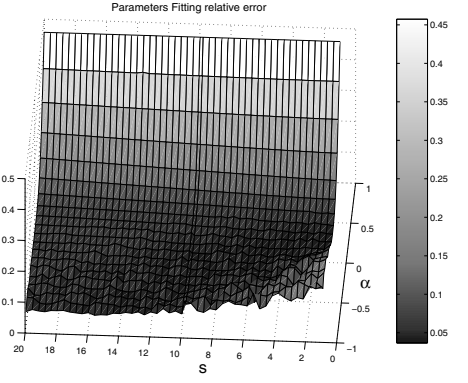



**Fig. 4.** Errors between the fit and the reference for different $\alpha$ and $s$. Notice the very broad ranges of $\alpha$ and $s$ that lead to a flat valley.

Observe how the valley along an $\alpha$ cross section is very large compared to a $s$ cross section. In general, it appears that $\alpha$ and $s$ can be chosen in a large range. Moreover, when $\alpha$ is set in the range of correct values, $s$ can vary within a very large range without really modifying the resulting fits. This robustness to the choice of parameters probably explains why robust fitting can be used with success in many practical applications.

## 3 Evaluation of the Detection

It is common knowledge that in practical applications, an estimate must be accompanied with some measure of *confidence* in order to be correctly used by higher level systems (e.g. Kalman filters in tracking applications [3]). In the statistical estimation framework, this confidence measure is naturally given by (the inverse of) the covariance matrix. This is why we have formulated detection as a fitting problem in the previous section. With Gaussian noise, computing the covariance matrix is straightforward. In the robust framework, however, a correct covariance matrix estimate is more difficult to obtain. There are two main reasons for that. First, there is no known closed-form solution, but only approximations. In this section, we focus on approximations that can efficiently be computed from the data in hand during the fitting. Second, it turns out that an accurate choice of the noise model parameters is much more critical for the covariance matrix estimation than for the fitting, as demonstrated below.

### 3.1 Confidence Matrices

A complete review of the many different approaches for approximating covariance matrices is out of the scope of this paper. Thus, we present a selection of five matrices that can be efficiently computed. Then, we propose a new approximation whose performances in terms of quality we will evaluate.

The simplest approximate covariance matrix $C_{Cipra}$, named Cipra's covariance matrix in [3], is:

$$C_{Cipra} = s^2 \left( \sum_{i=1}^{i=n} \lambda_i X_i X_i^t \right)^{-1},$$

where $\lambda_i$ are the weights when robust fitting converged. Another fast-to-compute approximation is proposed in [3]:

$$C_{simple} = s^2 \left( \sum_{i=1}^{i=n} \lambda_i^2 X_i X_i^t \right)^{-1}.$$

Notice that in these two estimates $s^2$ appears as a factor. As a consequence, incorrect values of the scale parameter lead to poor quality estimates.

In chapter 7 of [1], Huber derives an asymptotic covariance matrix and proposes three other approximate covariance matrices:

$$C_{Huber1} = K^2 \frac{\frac{1}{n-d-1} \sum_{i=1}^{i=n} (\rho'(b_i))^2}{(\frac{1}{n} \sum_{i=1}^{i=n} \rho''(b_i))^2} \left( \sum_{i=1}^{i=n} X_i X_i^t \right)^{-1},$$

$$C_{Huber2} = K \frac{\frac{1}{n-d-1} \sum_{i=1}^{i=n} (\rho'(b_i))^2}{\frac{1}{n} \sum_{i=1}^{i=n} \rho''(b_i)} \left( \sum_{i=1}^{i=n} \rho''(b_i) X_i X_i^t \right)^{-1},$$

$$C_{Huber3} = K^{-1} \frac{1}{n-d-1} \sum_{i=1}^{i=n} (\rho'(b_i))^2 W^{-1} \left( \sum_{i=1}^{i=n} X_i X_i^t \right) W^{-1},$$

with Huber's notation $\rho(t) = \phi(t^2)$ and $W = \sum_{i=1}^{i=n} \rho''(b_i) X_i X_i^t$. $n$ is the number of data points and $d+1$ is the dimension of vectors $A$ and $X_i$. The correction factor, $K$, is given by:

$$K = 1 + (d+1) \frac{\sum_{i=1}^{i=n} (\rho''(b_i) - \frac{1}{n} \sum_{i=1}^{i=n} \rho''(b_i))^2}{(\sum_{i=1}^{i=n} \rho''(b_i))^2}.$$

For easier notations, we set $O_1 = \sum_{i=1}^{i=n} \lambda_i X_i X_i^t$ and $O_2 = \sum_{i=1}^{i=n} \lambda_i^2 X_i X_i^t$. The inverse of a covariance matrix will be called a *confidence matrix* in the sequel.

In our experiments with real data, the previously proposed covariance matrices did not provide reliable estimations of the detected lane-markings. This is

experimentally shown in the next section. We propose the following new approximate covariance matrix which is justified in the Appendix:

$$C_{New} = \frac{\sum_{i=1}^{i=n} \lambda_i b_i^2}{\sum_{i=1}^{i=n} \lambda_i - Trace(O_2 O_1^{-1})} O_1^{-1} O_2 O_1^{-1}. \quad (4)$$

This approximation is not an asymptotic covariance matrix, unlike Huber's approximations. The advantage is thus that $C_{New}$ does not require a high number $n$ of data points to be applied. It relies on assuming non random weights $\lambda_i$. If the weights were considered as random variables, deriving the covariance matrix would become too involved.

### 3.2 Experimental Covariance Matrix Comparison

The six previous covariance matrix estimates are compared on average on 150 real images (partially shown in Fig. 3). The experimental comparison process is the same as in Sec. 2.4, with two additional steps:

5. The six approximate covariance matrices presented are computed for each image.
6. The reference matrix is computed as the covariance matrix of the fits. The relative errors between the reference and the six approximate covariance matrices are averaged on the image set for each $(\alpha, s)$.

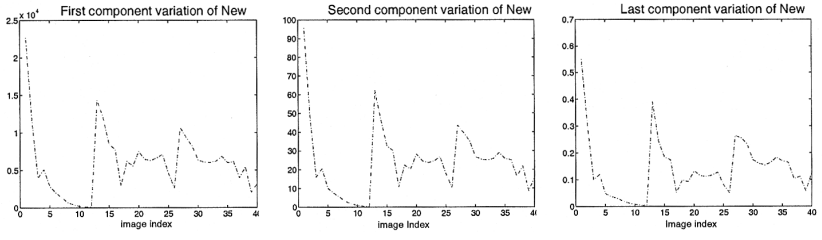



**Fig. 5.** These curves represent the variation of the three components of the $2 \times 2$ symmetric matrix $C_{New}$ versus the index of the image. The first and the last figures are associated to the diagonal components.

Fig. 5 illustrates the fact that the three components of a same matrix vary in the same way. Thus, without loss of generality, the analysis can be performed on only the first diagonal component.

Fig. 6 shows the value of the first component of the six confidence approximations for $\alpha = 0.2$ and $s = 2$. These variations are relatively similar, but the orders of magnitude can vary greatly from one approximation to another.

In Tab. 1, the relative errors with respect to the reference matrix for the different approximate covariance matrices are compared with $(\alpha, s) = (0.2, 2)$.

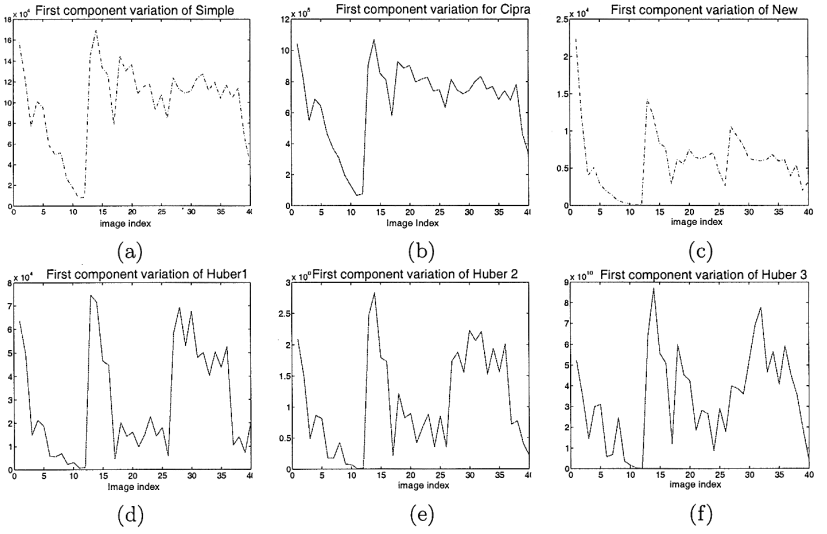



**Fig. 6.** These curves represent the first component of the different confidence matrices: $C_{Simple}$ (a), $C_{Cipra}$ (b), $C_{New}$ (c), $C_{Huber1}$ (d), $C_{Huber2}$ (e) and $C_{Huber3}$ (f). We notice that the values are of different orders of magnitude, but they vary in a similar way. This estimation is performed for $\alpha = 0.2$ and $s = 2$.

Similar results are obtained for other $(\alpha, s)$ in the valley of Fig. 4 where fits are accurate. The relative error is calculated with $\frac{2|C_{ij}-C_{ref,ij}|}{C_{ij}+C_{ref,ij}}$. Thus, 200% corresponds to the worst case where the value of $C_{ij}$ is far away from the reference value $C_{ref,ij}$. This table shows that only $C_{New}$ achieves a usable estimation of the confidence matrix. This is also illustrated by Fig. 6 where $\alpha = 0.2$ and $s = 2$ and where only $C_{New}$ has the correct order of magnitude. As a consequence, $C_{New}$ appears to be more robust than the other approximations.

In Fig. 7, the relative error between the first component of the confidence matrix $C_{New}$ and the reference is displayed for different values of $\alpha$ and $s$. For the other components, similar relative error maps are obtained. This relative error map shows that a relative error of 20% in the vicinity of $s = 2.5$ for $\alpha < 0$

**Table 1.** Relative errors in percentage on components $C_{11}$, $C_{22}$ and $C_{12}$ of the confidence matrix with respect to the reference, for the six covariance matrix estimates, for $(\alpha, s) = (0.2, 2)$.

| Type | $C_{11}$ rel.err. | $C_{22}$ rel.err. | $C_{12}$ rel.err. |
|---|---|---|---|
| Cipra | 195% | 198% | 196% |
| Huber1 | 156% | 183% | 192% |
| Huber2 | 200% | 200% | 200% |
| Huber3 | 200% | 200% | 200% |
| Simple | 98% | 99% | 101% |
| New | 64% | 41% | 43% |

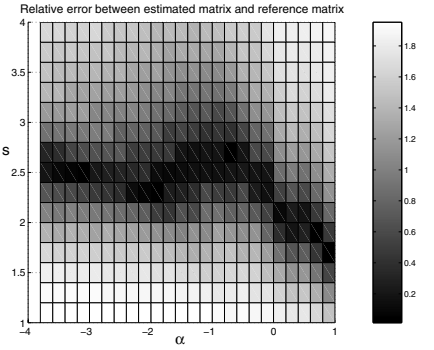



**Fig. 7.** Relative error of $C_{New}$ versus $(\alpha, s)$ with respect to reference covariance. With the new approximate covariance matrix, a relative error of 20% is achieved contrary to the others.

can be achieved. This was not obtained for the other approximations. This is due to the fact that $C_{New}$ is more robust to the choice of $\alpha$. Nevertheless, in contrast to parameter fitting, a correct estimate of $s$ remains important for a correct $C_{New}$.

In Sec. 2.3, the average power parameter $\alpha$ was estimated on the residual noise collected from 150 images as $\hat{\alpha} = 0.05$. Since the noise scale $s$ is not the same from one image to another, it is better to estimate the average scale as the average of the scales estimated on each image with $\alpha$ fixed to 0.05. The average model noise parameters obtained are $(\hat{\alpha}, \hat{s}) = (0.05, 2.02)$, which is in the valley observed in Fig. 7. Therefore, our experiments are consistent.

Similar synthetic experiments were performed for higher degree curves. Higher degree requires to introduce fitting regularisation. 10000 perturbed data sets with Cauchy noise were generated and fitted by a 2nd degree curve (3 parameters) assuming the true noise model. We obtained an average error of $-5\%$ for $C_{new}$, $+9\%$ for $C_{simple}$, $-16\%$ for $C_{cipra}$. Huber approximates are not of the same order of magnitude. This extends to higher degree results obtained on real data.

## 4 Conclusion

In this paper, we considered object detection as a statistical estimation problem. Then, the inverse covariance matrix naturally provides a measure of the confidence that can be associated to the detection result, and the non–Gaussian nature of the noise distributions that can be encountered in practice can be dealt with using robust fitting.

In this context, the important question which we have addressed is the *choice of the noise distribution model parameters*. We have experimentally shown that estimated fits remain relatively reliable even with a slightly incorrect noise model. On the contrary, correct covariance matrix estimates generally need fine tuning of the noise model parameters, which is difficult to achieve in practice.

We thus derived a new approximate covariance matrix (4) which is more robust to incorrect noise model parameters. This new approximate covariance matrix can be used to great advantage in many applications, including detection.

**Acknowledgement.** This work is supported by INRETS and région de l'Île de France. The authors are grateful to LIVIC technical team for its helps in the experiments.

## References

1. P. J. Huber. *Robust Statistics*. John Wiley and Sons, New York, New York, 1981.
2. T. Cipra and R. Romera. Robust kalman filtering and its application in time series analysis. *Kybernetika*, 27(6):481–494, 1991.
3. J.-P. Tarel, S.-S. Ieng, and P. Charbonnier. Using robust estimation algorithms for tracking explicit curves. In *European Conference on Computer Vision (ECCV'02)*, volume 1, pages 492–507, Copenhagen, Danmark, 2002.
4. S. Geman and D. McClure. Statistical methods for tomographic image reconstruction. *Bull. Int. Stat. Inst.*, LII(4):5–21, 1987.
5. J. Huang and D. Mumford. Statistics of natural images and models. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition (CVPR'99)*, pages 1541–1547, Ft. Collins, CO, USA, 23-25 June 1999.
6. R. Dahyot. *Appearance based road scene video analysis for management of the road network*. PhD thesis, University of Strasbourg I, 2001.
7. F.R. Hampel, P.J. Rousseeuw, E.M. Ronchetti, and W.A. Stahel. *Robust Statistics*. John Wiley and Sons, New York, New York, 1986.

## Appendix: $C_{New}$ Covariance Matrix Derivation

Let us justify $C_{New}$. The covariance matrix of the estimated parameter vector $\hat{A}$ is defined as the expectation $E[(\hat{A}-A)(\hat{A}-A)^t]$, where $A$ is the true parameter vector. We rewrite step 3 of the robust algorithm as $\hat{O_1}\hat{A} = \hat{B}$ with $\hat{B} = \sum_{i=1}^{i=n} \hat{\lambda_i}\hat{y_i}X_i$. For the true parameters, we have the similar equation $O_1 A = B$ where $B = \sum_{i=1}^{i=n} \lambda_i y_i X_i$. In the following, we remove the constraint that $\hat{\lambda_i}$ is a random variable and thus $\hat{\lambda_i} = \lambda_i$. This has three main consequences. First, it implies $\hat{O_1} = O_1$. Second, since $b_i$ is assumed centered, from $\hat{y_i} = X_i^t A + b_i$ we deduce $y_i = X_i^t A$, and thus $\hat{B} - B = \sum_{i=1}^{i=n} \lambda_i b_i X_i$. Since the $b_i$'s are independent random variables, this last equation implies $E[(\hat{B}-B)(\hat{B}-B)^t] = \sum_{i=1}^{i=n} \lambda_i^2 E[b_i^2] X_i X_i^t = E[b^2]O_2$. Third, using the previous equations, the covariance matrix can be approximated by $O_1^{-1}E[(B-\hat{B})(B-\hat{B})^t]O_1^{-t}$ or

$$C_{New} = E[b^2]O_1^{-1}O_2O_1^{-t}, \tag{5}$$

after substitution of the covariance matrix of $\hat{B}$. The point is now how to approximate the variance of the noise $E[b^2]$. The simplest idea is to compute it from

the noise model, but this leads to bad estimates when the noise model is not correctly chosen. A better idea, leading to estimates more robust to errors in the choice of the noise model, is to derive $E[b^2]$ from the data residuals $b_i$ using (3).

We can now derive the equation between $E[b^2]$ and $E[\hat{s}^2]$. (3) is rewritten as $\hat{s}^2 = \frac{1}{n}\sum_{i=1}^{i=n} \lambda_i(\hat{y_i} - X_i^t\hat{A})^2$. By introducing the true parameter vector $A$, inside the sum, we get $\hat{s}^2 = \frac{1}{n}\sum_{i=1}^{i=n} \lambda_i(b_i + X_i^t(A - \hat{A}))^2$. To calculate $E[\hat{s}^2]$ as a function of $E[b^2]$, we expand the squared term in the previous equation. Using (5), we deduce

$$E[\hat{s}^2] = \frac{1}{n}\sum_{i=1}^{i=n}(E[b^2]\lambda_i + E[b^2]\lambda_i X_i^t O_1^{-1} O_2 O_1^{-1} X_i + 2\lambda_i X_i^t E[(A - \hat{A})b_i]). \quad (6)$$

Since $b_i$ is centered, thus $E[(A-\hat{A})b_i] = -E[\hat{A}b_i]$ in (6). The estimated parameter vector obtained by fitting is $\hat{A} = O_1^{-1}\sum_{k=1}^{k=n} \lambda_k X_k \hat{y_k} = O_1^{-1}\sum_{i=1}^{k=n} \lambda_k X_k (X_k^t A + u_k)$. Therefore $E[\hat{A}b_i]$ can be expanded as $E[\hat{A}b_i] = O_1^{-1}\sum_{k=1}^{k=n} \lambda_k X_k E[u_k b_i]$. The independence of $b_i$ allows us to simplify the last equation in $E[\hat{A}b_i] = O_1^{-1}\lambda_i X_i E[b^2]$. As a consequence (6) is now:

$$E[\hat{s}^2] = \frac{E[b^2]}{n}(\sum_{i=1}^{i=n}\lambda_i + \sum_{i=1}^{i=n}\lambda_i X_i^t O_1^{-1} O_2 O_1^{-1} X_i - 2\sum_{i=1}^{i=n}\lambda_i^2 X_i^t O_1^{-1} X_i). \quad (7)$$

Using the property $Trace(AB) = Trace(BA)$, we can rewrite the second and third terms in (7) as $\sum_{i=1}^{i=n}\lambda_i X_i^t O_1^{-1} O_2 O_1^{-1} X_i = \sum_{i=1}^{i=n}\lambda_i^2 X_i^t O_1^{-1} X_i = Trace(O_2 O_1^{-1})$. This implies the simple result $E[\hat{s}^2] = \frac{E[b^2]}{n}\left(\sum_{i=1}^{i=n}\lambda_i - Trace(O_2 O_1^{-1})\right)$. Using (3) as an approximate value of $E[\hat{s}^2]$, we deduce the following estimate of $E[b^2]$:

$$E[b^2] = \frac{\sum_{i=1}^{i=n}\lambda_i b_i^2}{\sum_{i=1}^{i=n}\lambda_i - Trace(O_2 O_1^{-1})}. \quad (8)$$

The substitution of (8) in (5), results in the proposed approximate covariance matrix (4).

# Local Orientation Smoothness Prior for Vascular Segmentation of Angiography

Wilbur C.K. Wong[1], Albert C.S. Chung[1], and Simon C.H. Yu[2]

[1] Department of Computer Science,
The Hong Kong University of Science and Technology,
Clear Water Bay, Kowloon, HK
{cswilbur,achung}@cs.ust.hk
[2] Department of Diagnostic Radiology and Organ Imaging,
The Prince of Wales Hospital, Shatin, NT, HK
simonyu@cuhk.edu.hk

**Abstract.** We present a new generic method for vascular segmentation of angiography. Angiography is used for the medical diagnosis of arterial diseases. To facilitate an effective and efficient review of the vascular information in the angiograms, segmentation is a first stage for other post-processing routines. The method we propose uses a novel *a priori* — local orientation smoothness prior — to enforce an adaptive regularization constraint for the vascular segmentation within the Bayes' framework. It aspires to segment a variety of angiographies and is aimed at improving the quality of segmentation in low blood flow regions. Our algorithm is tested on numerical phantoms and clinical datasets. The experimental results show that our method produces better segmentations than the maximum likelihood estimation and the estimation with a multi-level logistic Markov random field model. Furthermore, the novel algorithm produces aneurysm segmentations comparable to the manual segmentations obtained from an experienced consultant radiologist.

## 1 Introduction

Evolution in technology surrounding vascular imaging has brought radiologists to non-invasive imaging modalities that can provide accurate 3D vascular information quickly to a variety of patients. Vascular imaging helps the physician to define the character and extent of a vascular disease, thereby aiding diagnosis and prognosis. To facilitate an effective and efficient review of the vascular information in angiograms, segmentation is a first stage for other post-processing routines or analyses, such as visualization, volumetric measurement, quantitative comparison and image-guided surgery [1].

A variety of approaches have been proposed for vascular segmentation. For instance, authors in [2], [3] demonstrated that the expectation maximization (EM) algorithm and a maximum likelihood (ML) estimate can be used to segment vascular structures automatically with a proper statistical mixture model. The use of gradient information to drive evolving contours with the level set

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 353–365, 2004.
© Springer-Verlag Berlin Heidelberg 2004

method and topologically adaptable surfaces to segment vasculature in the angiograms has been proposed in [4], [5], [6], [7]. Region-growing approaches to segmenting the angiograms with initial segmentations have been illustrated in [8], [9]. Alternatively, a tubular object has been used to model vessel segment in the angiograms explicitly for ridge transversal, multiscale analysis of vasculature and vessel diameter estimation [10], [11]. A multiscale line enhancement filter has been applied to segment curvilinear structures in the angiograms [12].

In a previous published work [3], we demonstrated a method to combine speed and phase information for vascular segmentation of phase-contrast (PC) magnetic resonance angiography[1] (MRA). Since phase images are only available in PC MRA, we introduce a novel generic framework for vascular segmentation of a variety of angiographies. The new method depends solely on a speed image to segment PC MRA and aspires to segment other angiographies such as time-of-flight (TOF) MRA and 3D rotational angiography (RA).

When blood flows along a vessel, because of the blood viscosity, frictional force slows down the flow near the vascular wall [13]. As such, the intensity values are low at the boundary of vessels in the angiograms. The inhomogeneous regions are a challenge if vascular segmentation is to be robust.

The method we propose uses a new smoothness prior to improve the quality of segmentation, particularly, at the low blood flow regions. The *a priori*, namely *local orientation smoothness prior*, exploits local orientation smoothness of the vascular structures to enforce an adaptive regularization constraint for robust vascular segmentation.

We expect the application of the *a priori* can be extended to different areas, e.g., image restoration with edge-preserving or coherent-enhancing capability (see [14], [15] and references therein), segmentation of non-medical images such as radar images [16] and object extraction from video [17].

In the next section, we present the Bayes' approach to segmenting the angiograms. We describe a robust method used to estimate local orientation from the images in Section 3. The implementation issues are outlined in Section 4. The experimental results on numerical phantoms and clinical datasets are given in Section 5, and conclusions are drawn in Section 6.

## 2 Bayes' Approach to Segmenting Angiograms

In this section, we formulate the vascular segmentation problem on the Bayes' framework. We discuss the estimation of the global likelihood probability and present the definition of the new *a priori* used to enforce an adaptive regularization constraint for the segmentation.

### 2.1 Problem Formulation

A vascular segmentation problem is regarded as a process to assign labels from a label set $\mathcal{L} = \{vessel, background\}$ to each of the voxels indexed in

[1] Magnetic resonance angiography is one of most widely available vascular imaging techniques in a clinical environment.

$\mathcal{S} = \{1, \ldots, m\}$, where $m$ is the total number of voxels in an angiogram $\boldsymbol{y}$. Let a vector $\boldsymbol{x}$ be a segmentation of the image $\boldsymbol{y}$, each element in the vector $\boldsymbol{x}$ can be regarded as a mapping from $\mathcal{S}$ to $\mathcal{L}$, i.e., $x_i : \mathcal{S} \rightarrow \mathcal{L}$. A feasible segmentation $\boldsymbol{x}$ is, therefore, in a Cartesian product $\Omega_x$ of the $m$ label sets $\mathcal{L}$. The set $\Omega_x$ is known as a *configuration space*. In the Bayes' framework, the optimal solution is given by a feasible segmentation $\boldsymbol{x}^*$ of the angiogram $\boldsymbol{y}$, which maximizes *a posteriori* probability $p(\boldsymbol{x} \mid \boldsymbol{y}) \propto p(\boldsymbol{y} \mid \boldsymbol{x})\, p(\boldsymbol{x})$ over the space $\Omega_x$ [18]. The likelihood probability $p(\boldsymbol{y} \mid \boldsymbol{x})$ can be application-specific. It suggests the likelihood of a particular label assignment, based on the intensity values in the angiogram $\boldsymbol{y}$. Whereas, the prior probability $p(\boldsymbol{x})$ constrains the solution contextually.

In order to have a tractable constraint, the Markov random field (MRF) theory is used. By virtue of the Hammersley-Clifford theorem [19], the Gibbs distribution provides us with a practical way of specifying the joint probability — the prior probability $p(\boldsymbol{x})$ — of an MRF. The maximum *a posteriori* (MAP) estimate $\boldsymbol{x}^*$, therefore, becomes a minimum of the summation of the likelihood energy and prior energy functions over the configuration space $\Omega_x$,

$$\boldsymbol{x}^* = \arg \min_{\boldsymbol{x} \in \Omega_x} \Big( U(\boldsymbol{y} \mid \boldsymbol{x}) + U(\boldsymbol{x}) \Big), \tag{1}$$

where $U(\boldsymbol{y} \mid \boldsymbol{x}) = -\log p(\boldsymbol{y} \mid \boldsymbol{x})$ is the likelihood energy function; and $U(\boldsymbol{x}) = \sum_{c \in \mathcal{C}} V_c(\boldsymbol{x})$ is the prior energy function, which is a sum of *clique potentials* $V_c(\boldsymbol{x})$ over all possible cliques in $\mathcal{C} \subseteq \mathcal{S}$ [18].

### 2.2 Estimation of the Global Likelihood Probability

In practice, because of the high complexity of the random variables $\boldsymbol{x}$ and $\boldsymbol{y}$, it is computationally intractable to calculate the likelihood energy $U(\boldsymbol{y} \mid \boldsymbol{x})$ from the negative log-likelihood, $-\log p(\boldsymbol{y} \mid \boldsymbol{x})$. As such, we assume that the intensity value of each voxel $y_i$ is independent and identically distributed (i.i.d.). The calculation of the likelihood energy becomes tractable since the global likelihood probability can be determined by local likelihood probabilities and the likelihood energy function can be expressed as:

$$U(\boldsymbol{y} \mid \boldsymbol{x}) = -\sum_{i \in S} \log p(y_i \mid x_i). \tag{2}$$

### 2.3 Local Orientation Smoothness Prior

The new *a priori* is presented in this section. It exploits local orientation smoothness of the vascular structures and is used to enforce an adaptive regularization constraint for the vascular segmentation within the Bayes' framework. A smoothness constraint has been used to solve low level vision problems. Applications such as surface reconstruction, optical flow determination and shape extraction, demonstrate that this generic contextual constraint is a useful *a priori* to a variety of low level vision problems [18]. In the MRF framework, contextual

constraint is expressed as the prior probability or the equivalent prior energy function $U(\boldsymbol{x})$ as given in Equation 1.

Because of the blood viscosity, blood flows are low at the boundary of vessels [13]. Using speed information alone cannot give satisfactory segmentation in low blood flow regions [3]. Therefore, in this paper, we propose a local orientation smoothness prior, aiming at improving the quality of segmentation at the low blood flow regions. The local orientation smoothness prior is expressed as follows:

$$U(\boldsymbol{x}) = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i} \Big(1 - f(x_i)\Big) f(x_j)\, g(i,j) \Big(\beta_1 h_1(i,j) + \beta_2 h_2(i,j)\Big), \quad (3)$$

where $\mathcal{N}_i$ denotes a set of voxels neighboring the voxel $i$ with respect to a neighborhood system $\mathcal{N} \subseteq \mathcal{S}$; $f$ is a mapping function defined as follows:

$$f(x_i) = \begin{cases} 0, x_i = \textit{background}, \\ 1, x_i = \textit{vessel}; \end{cases} \quad (4)$$

$g(i,j)$ measures the geometric closeness (Euclidean distance) between voxels $i$ and $j$; $h_1(i,j)$ and $h_2(i,j)$ measure the linear (rank 1) and planar (rank 2) orientation similarities at voxel $i$ in respect of voxel $j$ respectively; $\beta_1$ and $\beta_2$ are positive weights, which need not sum to one and are used to control the influence of orientation smoothness in the interactions between the adjacent voxels. The idea of applying geometric closeness and similarity measures as constraints is similar to the one exploited in the bilateral filters [20]. In other words, the function $g$ in Equation 3 defines the structural locality, whereas the functions $h_1$ and $h_2$ quantify the structural orientation smoothness.

In this paper, the geometric closeness, $g$, and orientation similarity measures, $h_1$ and $h_2$, are Gaussian functions of the magnitude of the relative position vector of voxel $j$ from voxel $i$, $\|\boldsymbol{u}_{ij}\|$, and the orientation discrepancy, $\delta$, between voxels $i$ and $j$ respectively. The geometric closeness function is given as a decreasing function $g$ when the distance $\|\boldsymbol{u}_{ij}\|$ increases:

$$g(i,j) = \exp\left(-\frac{\|\boldsymbol{u}_{ij}\|^2}{2\sigma_g^2}\right), \quad (5)$$

where $\boldsymbol{u}_{ij}$ is the relative position vector of voxel $j$ from voxel $i$ and the parameter $\sigma_g$ defines the desired geometric influence between neighboring voxels. The orientation similarity function $h_k$ is written as a decreasing function when the orientation discrepancy $\delta$ increases:

$$h_k(i,j) = \exp\left(-\frac{\delta^2(\hat{\boldsymbol{u}}_{ij}, \hat{\boldsymbol{w}}_k)}{2\sigma_h^2}\right), \quad (6)$$

where $k = 1$ and $k = 2$ denote rank 1 and rank 2 orientation similarities respectively; the discrepancy function $\delta$ is defined as follows:

$$\delta(\boldsymbol{u}, \boldsymbol{v}) = 1 - \left|\boldsymbol{u}^T \boldsymbol{v}\right|; \quad (7)$$

$\hat{\boldsymbol{w}}_1$ depicts the principal direction of a linear orientation, while $\hat{\boldsymbol{w}}_2$ depicts one of the principal directions of a planar orientation[2]; and the parameter $\sigma_h$ is chosen based on the desired amount of orientation discrepancy filtering amongst adjacent voxels.

In other words, the prior energy function in Equation 3 encourages piecewise continuous vessel label assignment in the segmentation. Vascular piecewise continuity is constrained by geometric closeness and orientation similarity measures. As long as voxels $i$ and $j$ are close enough, with similar rank 1 and/or rank 2 orientations, and the label assigned to voxel $j$ is *vessel*, i.e., $f(x_j) = 1$, it is in favor of vessel label assignment to voxel $i$, i.e., $f(x_i) = 1$. This is because we are minimizing the energy function in Equation 1. On the other hand, if the label assigned to voxel $j$ is *background*, i.e., $f(x_j) = 0$, the prior energy vanishes and the label assignment to voxel $i$ is based solely on the likelihood energy.

## 3 Estimating Local Orientation by Eigen Decomposition of Orientation Tensor

Recall from Section 2 that we exploit the principal directions of linear and planar orientations, $\hat{\boldsymbol{w}}_1$ and $\hat{\boldsymbol{w}}_2$, to constrain the segmentation with the local orientation smoothness prior. In this section, we describe a robust method to estimate the principal directions. The estimation is obtained by an orientation tensor rather than a conventional Hessian matrix for the robustness to noise and performance reasons, see Section 5.1, for the performance comparisons of the two methods.

### 3.1 Orientation Tensor

The use of an orientation tensor for local structure description was first presented in Knutsson's work [21], which was motivated by the need to find a continuous representation of local orientation. Knutsson formulated the orientation tensor by combining outputs from a number of directional polar separable quadrature filters. A quadrature filter is constructed in the Fourier domain. It is a complex valued filter in the spatial domain, which can be viewed as a pair of filters: (1) symmetric (line filter) and (2) antisymmetric (edge filter). Further, it is orientation-specific and is sensitive to lines and edges that are orientated at the filter direction. In Knutsson's formulation, the orientation tensor $\mathbf{T}$ in a 3D space is defined as:

$$\mathbf{T} = \sum_{k=1}^{6} q_k \left( \frac{5}{4} \hat{\boldsymbol{n}}_k \hat{\boldsymbol{n}}_k^T - \frac{1}{4} \mathbf{I} \right), \tag{8}$$

where $q_k$ is the modulus of the complex valued response from a quadrature filter in the direction $\hat{\boldsymbol{n}}_k$ and $\mathbf{I}$ is the identity tensor. For further details see [21] or Chapter 6 in [22].

[2] The orientation of a planar structure is depicted by two principal directions, which are orthogonal to its normal vector; as a planar structure can be seen as a series of linear structures, it is noted that, other than the vector $\hat{\boldsymbol{w}}_2$, $\hat{\boldsymbol{w}}_1$ depicts the other principal direction of the planar structure.

### 3.2 Estimating Local Orientation

Estimation of local orientation is performed via eigen decomposition of the orientation tensor $\mathbf{T}$ at each voxel in an image [22]. To calculate the tensor $\mathbf{T}$, the image should be convolved with the six quadrature filters. After the convolutions, there are six moduli of the complex valued filter responses associated with each voxel, $q_k$, $k = 1, 2, \ldots 6$. The tensor is computed as stated in Equation 8.

Let $\lambda_1$, $\lambda_2$ and $\lambda_3$ be the eigenvalues of the tensor $\mathbf{T}$ in descending order ($\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$) and $\hat{\boldsymbol{e}}_i$ ($i = 1, 2, 3$) are the corresponding eigenvectors respectively. The estimation of the local orientation can be one of the three cases as follows: (a) planar case: $\lambda_1 \gg \lambda_2 \simeq \lambda_3$, $\hat{\boldsymbol{e}}_2$ and $\hat{\boldsymbol{e}}_3$ are estimates to the principal directions of the planar structure; (b) linear case: $\lambda_1 \simeq \lambda_2 \gg \lambda_3$, $\hat{\boldsymbol{e}}_3$ is an estimate to the principal direction of the linear structure; and (c) isotropic case: $\lambda_1 \simeq \lambda_2 \simeq \lambda_3$, no specific orientation. Therefore, we can approximate the principal directions of the linear and planar orientations ($\hat{\boldsymbol{w}}_1$ and $\hat{\boldsymbol{w}}_2$ in Equation 6) with the eigenvectors $\hat{\boldsymbol{e}}_3$ and $\hat{\boldsymbol{e}}_2$ respectively.

## 4 Implementation

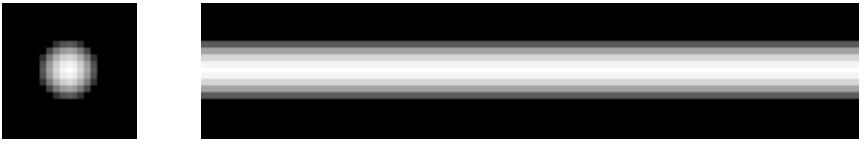
The proposed algorithm, summarized in Algorithm 1, consists of three parts. We discuss each of the three parts in Sections 4.1, 4.2 and 4.3 respectively.

**Algorithm 1** Main algorithm

1: Estimate local orientation with an orientation tensor, compute $\hat{\boldsymbol{e}}_3$ and $\hat{\boldsymbol{e}}_2$ (i.e., $\hat{\boldsymbol{w}}_1$ and $\hat{\boldsymbol{w}}_2$) and the likelihood probability $p\left(y_i \mid x_i\right)$ at each voxel
2: Initialize the algorithm with an ML estimate $\boldsymbol{x}^0$, $k \Leftarrow 0$
3: **repeat**
4: $\quad k \Leftarrow k + 1$
5: $\quad$ **for all** $i$ in the set $\mathcal{S}$ **do**
6: $\quad\quad E_v \Leftarrow -\log p\left(y_i \mid \mathit{vessel}\right)$
7: $\quad\quad E_b \Leftarrow \sum_{j \in \mathcal{N}_i} f\left(x_j^{k-1}\right) g\left(i,j\right) \left(\beta_1 h_1\left(i,j\right) + \beta_2 h_2\left(i,j\right)\right) - \log p\left(y_i \mid \mathit{background}\right)$
8: $\quad\quad$ **if** $E_v > E_b$ **then**
9: $\quad\quad\quad x_i^k \Leftarrow \mathit{background}$
10: $\quad\quad$ **else**
11: $\quad\quad\quad x_i^k \Leftarrow \mathit{vessel}$
12: **until** convergence
13: Return the final segmentation $\boldsymbol{x}^k$

### 4.1 Estimating Local Orientation

To estimate the local orientation with an orientation tensor, as outlined in Section 3, six quadrature filters of a window size $5 \times 5 \times 5$, relative bandwidth $B$ equals 2 and center frequency $\rho$ equals $\frac{\pi}{2\sqrt{2}}$ are used (relative bandwidth $B$ and



(a) (b)

**Fig. 1.** Horizontal pipe phantom with a parabolic flow profile. (a) Cross section orthogonal to the pipe orientation, (b) central cross section along the pipe orientation

center frequency $\rho$ control the characteristics of the quadrature filters, see [22] for further details). The 3D image is convolved with the filters to obtain six moduli of complex valued responses at each voxel. Then the orientation tensor is computed as given in Equation 8. The eigen decomposition of the orientation tensor is performed and the two vectors $\hat{\boldsymbol{e}}_3$ and $\hat{\boldsymbol{e}}_2$, which depict the principal directions of the linear and planar orientations, are obtained at each voxel.

### 4.2 Likelihood Estimation and Algorithm Initialization

As discussed in Section 2.1, the likelihood estimation can be application-specific. In this work, we use statistical mixture models to estimate the likelihood probabilities of the numerical phantoms and clinical datasets under the i.i.d. assumption (details are given in Sections 5.2 and 5.3 respectively).

To initialize the algorithm, an ML estimate is used. Given the likelihood probabilities are known, the initial segmentation $\boldsymbol{x}^0$ is obtained as follows:

$$\boldsymbol{x}^0 = \left\{ \arg\max_{x_i \in \mathcal{L}} p\left(y_i \,|\, x_i\right) \,|\, \forall i \in \mathcal{S} \right\}. \tag{9}$$

### 4.3 Solution by Iterated Conditional Modes

We use iterated conditional modes (ICM) [23] to solve the minimization problem in Equation 1 with deterministic local search for the following reasons: (1) the formulations of the likelihood energy and prior energy functions are entirely based on local information, see Sections 2.2 and 2.3 for details respectively; (2) our initial estimate of the truth segmentation can be very close to the optimal solution; and (3) the ICM optimization algorithm gives fast convergence to the solution and is simple to implement, which make it more attractive to time-critical medical applications than other optimization algorithms.

## 5 Validation, Sensitivity Analyses, and Experiments

In this section, the local orientation estimation (described in Section 3) is validated. Furthermore, sensitivity of the proposed algorithm is studied and experiments on numerical phantoms and clinical datasets are presented.

(a) Without noise (b) With noise

**Fig. 2.** Orientation discrepancies between the estimated and the truth flow iso-surface normals. (a) Without noise; (b) with noise, the orientation tensor approach only

### 5.1 Validation of Local Orientation Estimation

As mentioned in Section 3, using an orientation tensor is not the only approach to estimating local orientation. A Hessian matrix (defined as in [12]) can also be used for the estimation (see [10], [11]). In this section, we have conducted experiments to compare the performance of the two approaches.

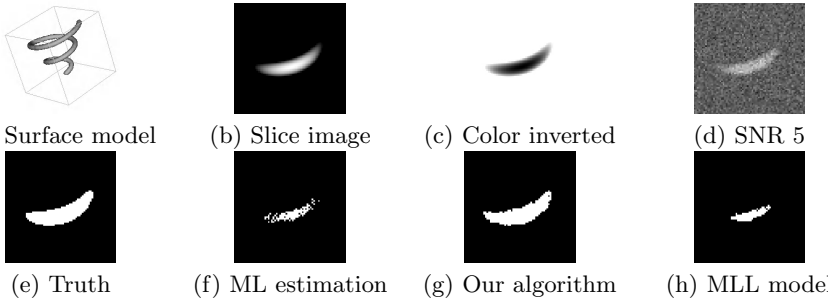
A numerical horizontal pipe phantom with a parabolic flow profile[3] (peak flow magnitude equals 255) in a volume of size $100 \times 21 \times 21$ voxels has been built. The diameter of the pipe is 9 voxels, which is the average diameter of the major brain vessels in the clinical datasets. Figure 1 shows the noiseless pipe phantom.

We have compared the performance of the orientation tensor and Hessian matrix approaches (hereafter referred to as "OT" and "HESSIAN" respectively) on the noiseless phantom. Comparison is based on the orientation discrepancy (function $\delta$ in Equation 7) between the estimated and the truth flow iso-surface normals[4]. In OT, a $5 \times 5 \times 5$ filter window with relative bandwidth $B$ equals 2 and center frequency $\rho$ equals $\frac{\pi}{2\sqrt{2}}$ has been used; a $3 \times 3$ Gaussian kernel with $\sigma = 1$ has been employed for tensor averaging (for further details, see Chapter 6 in [22]). While a $5 \times 5 \times 5$ Gaussian smoothing kernel with $\sigma = \frac{5}{3}$ and a central finite difference approximation have been used in HESSIAN. Figure 2(a) shows the comparison between the two approaches. It is evident that OT gives a close-to-prefect orientation estimation. Conversely, owing to the use of second order derivatives and finite difference approximations, HESSIAN produces less than satisfactory results.

Furthermore, OT has been evaluated at different levels of additive white Gaussian noise. Signal-to-noise ratio (SNR) is defined as the ratio of the peak intensity value to the sample standard deviation of the noise. Figure 2(b) shows the discrepancy measures amongst SNR 2, 5 and 10 (on average, SNR is found to be about 5 in the clinical datasets). It is noted that OT is robust to noise. It gives, on average, a discrepancy value 0.05, i.e., 18° deviation from the truth flow iso-surface normals, even in the phantom corrupted by severe noise.

[3] A parabolic flow model is the simplest model to study blood flow in vessels [13].

[4] We may think of the axis-symmetric flow in the pipe phantom as the sliding of a series of concentric tubes of fluid [13]. The flow iso-surface normals are referred to as the surface normals of these tubes.



(a) Surface model (b) Slice image (c) Color inverted (d) SNR 5

(e) Truth (f) ML estimation (g) Our algorithm (h) MLL model

**Fig. 3.** Pipe phantom (radius equals 5 voxels) with the centerline aligned with a cubic B-spline curve in a 3D space. (a) 3D surface model, (b) a portion of a slice image, (c) color inverted slice image, (d) slice image with SNR equals 5, (e) truth segmentation, (f) ML estimation, (g) our estimation, (h) estimation with a MLL MRF model

### 5.2 Sensitivity Analyses and Experiments on Numerical Phantoms

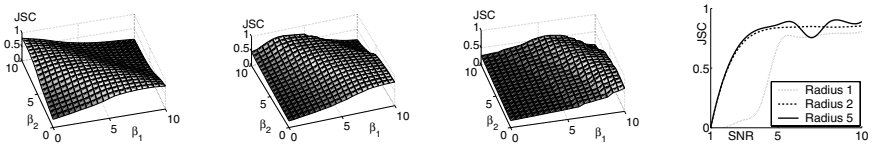
There are four free parameters in the proposed algorithm, they are: $\sigma_g$ in Equation 5, $\sigma_h$ in Equation 6, $\beta_1$ and $\beta_2$ in Equation 3. The parameters $\sigma_g$ and $\sigma_h$ define the desired geometric influence and amount of orientation discrepancy filtering amongst neighboring voxels respectively. Whereas, the two positive weights $\beta_1$ and $\beta_2$ control the influence of orientation smoothness in the interactions between the adjacent voxels.

Plausible values of the parameters $\sigma_g$ and $\sigma_h$ are suggested in this paragraph. To compromise between computational speed and robustness of the algorithm, a $3 \times 3 \times 3$ neighborhood system is used in the ICM algorithm. This leads to a justifiable choice to set $\sigma_g = 1$. For the orientation discrepancy filtering, we suggest $\sigma_h = 0.2$, this implies the algorithm has a 95% cut-off at discrepancy measure equals $2\sigma_h = 0.4$. In other words, the algorithm filters out neighboring voxels that are located outside the capture range of the filter, $\pm 53^\circ$ deviation from the estimated orientations, $\hat{\boldsymbol{w}}_1$ and $\hat{\boldsymbol{w}}_2$.

To understand the relationships between the two positive weights $\beta_1$ and $\beta_2$ and the sensitivity of the algorithm towards different object sizes and noise levels, a numerical phantom with a parabolic flow profile has been built. It is a pipe with the centerline aligned with a cubic B-spline curve in a 3D space. The phantom volume size is $128 \times 128 \times 128$ voxels. Figure 3(a) shows the phantom as a 3D surface model.

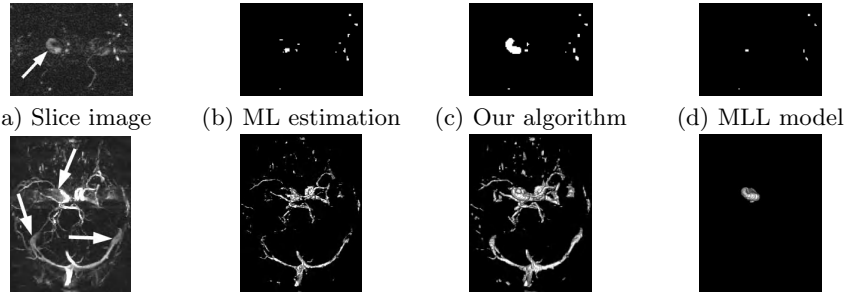
The relationships between $\beta_1$ and $\beta_2$ with pipes, corrupted by additive white Gaussian noise, in different radii, 5, 2 and 1 voxel(s) (in the clinical datasets, vessel radius ranges from 1 voxel to 5 voxels) are shown in Figures 4(a), 4(b) and 4(c) respectively[5]. The SNR of the pipe phantoms is 5. SNR is defined as in Section 5.1. The vertical axis of the graphs shows the Jaccard similarity coefficient (JSC) between the estimated and the truth segmentations. JSC is defined as the ratio of the intersection volume to the union volume of the two

[5] A Gaussian-uniform mixture model is used to estimate the likelihood probabilities.




(a) 5 voxels radius (b) 2 voxels radius (c) 1 voxel radius (d) Noise sensitivity

**Fig. 4.** Graphs from sensitivity analyses. (a)-(c) Relationship between $\beta_1$ and $\beta_2$, and (d) noise sensitivity of the algorithm with pipes in different radii



(a) Slice image (b) ML estimation (c) Our algorithm (d) MLL model

(e) MIP image (f) ML estimation (g) Our algorithm (h) Manual

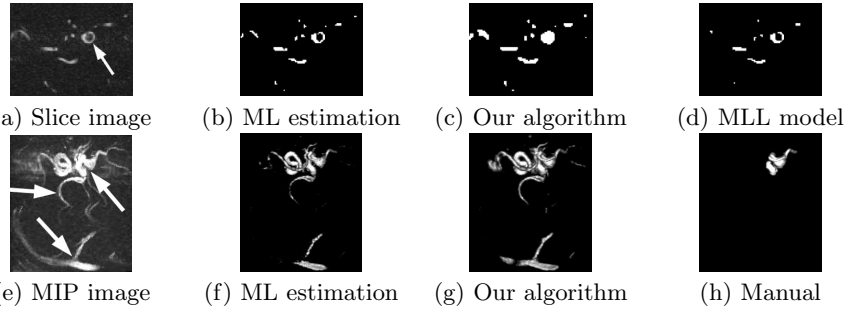
**Fig. 5.** PC MRA dataset 1. (a) A slice image from the dataset; segmentation with (b) the ML estimation, (c) our algorithm, (d) the estimation with an MLL MRF model; (e) maximum intensity projection (MIP) image; volume rendered image of (f) segmentation with the ML estimation, (g) segmentation with our algorithm, (h) manual segmentation of the subregion that contains an aneurysm

given segmentations [24]. It is a similarity measure that maintains a balance between the sensitivity and specificity, and is used to quantify the accuracy of an estimated segmentation. JSC gives value 1 if the estimated segmentation equals the truth segmentation.

From the figures, it is noted that $\beta_1$ and $\beta_2$ complement each other. As the radius of the pipe decreases, $\beta_1$ contributes more to a better estimation; to recapitulate, $\beta_1$ controls the influence of the linear orientation smoothness in the interactions amongst neighboring voxels.

Figure 4(d) shows the noise sensitivity analysis of the algorithm with pipes in different radii. The weights $\beta_1$ and $\beta_2$ are set to the values that give maximum JSC as shown in Figures 4(a)–(c) for different pipe radii. It is shown that the algorithm is robust to noise over a wide range of object sizes. For small objects, i.e. 1 voxel in radius, the algorithm can give a satisfactory estimation if SNR $\geq 5$ (The average SNR in the clinical datasets is found to be 5).

Figures 3(f), 3(g) and 3(h) show the segmentations obtained with the ML estimation, our algorithm and the estimation with a multi-level logistic (MLL) MRF model respectively. It is indicated that our algorithm produces satisfactory segmentation of the pipe (JSC = 0.88) contrary to the ML estimation (JSC = 0.29) and the estimation with the MLL MRF model (JSC = 0.26).



(a) Slice image (b) ML estimation (c) Our algorithm (d) MLL model

(e) MIP image (f) ML estimation (g) Our algorithm (h) Manual

**Fig. 6.** PC MRA dataset 2. Figure arrangement is identical to Figure 5

### 5.3 Experiments on Clinical Datasets

The proposed segmentation algorithm has been applied to two clinical datasets. They were obtained from the Department of Diagnostic Radiology and Organ Imaging at the Prince of Wales Hospital, Hong Kong. The two PC MRA intracranial scans were acquired from a Siemens 1.5T Sonata imager. The data volume is $256 \times 176 \times 30$ voxels with a voxel size of $0.9 \times 0.9 \times 1.5\text{mm}^3$.

The experimental results are shown in Figures 5 and 6. It is evident that parts of the vascular structures in the low blood flow regions, i.e., the boundary of vessels and the aneurysms[6] (indicated by the arrows), are left out in the segmentations produced by the ML estimation and the estimation with the MLL MRF model. On the contrary, our algorithm gives satisfactory segmentations in those regions.

Furthermore, our method produces aneurysm segmentations comparable to the manual segmentations obtained from a local consultant radiologist who has $> 15$ years' clinical experience in endovascular neurosurgery (on average, JSC $= 0.73$ for our algorithm versus JSC $= 0.41$ for the ML estimation[7]). See Figures 5(h) and 6(h) for the volume rendered images[8] of the manual segmentations for comparisons. It is observed that there is a large improvement in segmentation using our algorithm, especially for the vessels with low intensity values.

One may find that a few small vessels are left out in the segmentations produced by our algorithm showed in Figures 5(g) and 6(g). According to the radiologists' feedback, the segmentations obtained with our algorithm are good enough for clinical applications. Small vessels with diameter $< 3$ voxels are not their current primary interest in this work.

In the experiments, the same set of parameter values is used for the segmentations of different clinical datasets. The parameter values are: $\sigma_g = 1$, $\sigma_h = 0.2$ and $\beta_1 = \beta_2 = 3.5$. The Maxwell-Gaussian-uniform (MGU) mixture model is

[6] An aneurysm is an abnormal local dilatation of blood vessel.

[7] The JSC values are computed within the subregions that contain the aneurysms with the manual segmentations treated as the truth segmentations.

[8] Volume rendering is performed using the Visualization Toolkit (`www.vtk.org`).

employed to estimate the likelihood probabilities as suggested in [3]. On average, the algorithm takes 42s, needs $< 20$ iterations to converge and consumes $< 100$MB of memory to segment the two PC MRA datasets on a 2.66GHz PC.

## 6 Conclusions

We have presented a new generic method for vascular segmentation of angiography. Our method uses a novel smoothness prior that exploits local orientation smoothness of the vascular structures to improve the quality of segmentation at low blood flow regions. The *a priori* is expressed as a function of geometric closeness and orientation similarity measures. Furthermore, we have described a method to estimate the local orientation with an orientation tensor. The experimental results have indicated that the method is more robust than that with a conventional Hessian matrix, and is capable of a more accurate orientation estimation.

Our algorithm has been applied to the numerical phantoms and clinical datasets. The experimental results have shown that the new method produces segmentations better than the maximum likelihood estimation and the estimation with the multi-level logistic Markov random field model. Moreover, the aneurysm segmentations obtained with our method are comparable to the manual segmentations produced by an experienced consultant radiologist.

In this work, we have demonstrated an application of the local orientation smoothness prior to medical image segmentation. We expect the use of the *a priori* to be applicable to different areas such as image restoration with edge-preserving or coherent-enhancing capability, non-medical image segmentation and object extraction from video.

**Acknowledgments.** This work is supported in part by the Hong Kong Research Grants Council (HK RGC) under grants HKUST6209/02E and DAG01/02.EG04, and the Sino Software Research Institute (SSRI) under grant SSRI01/02.EG22. The authors wish to thank Prof. C.-F. Westin for the fruitful discussions, and providing MATLAB codes on the construction of the quadrature filters at the early stage of the software development. They would also like to thank Craig C. W. Jo and the anonymous reviewers for their perceptive comments, which have significantly improved the paper.

## References

1. Suri, J.S., Liu, K., Reden, L., Laxminarayan, S.: A review on MR vascular image processing algorithms: Acquisition and prefiltering: Part I. IEEE Trans. Inform. Technol. Biomed. **6** (2002) 324–337
2. Wilson, D.L., Noble, J.A.: An adaptive segmentation algorithm for time-of-flight MRA data. IEEE Trans. Med. Imag. **18** (1999) 938–945
3. Chung, A.C.S., Noble, J.A., Summers, P.: Fusing speed and phase information for vascular segmentation of phase contrast MR angiograms. MedIA **6** (2002) 109–128

4. Lorigo, L.M., Faugeras, O., Grimson, W.E.L., Keriven, R., Kikinis, R., Westin, C.F.: Co-dimension 2 geodesic active contours for MRA segmentation. In: IPMI. (1999) 126–139
5. Wang, K.C., Dutton, R.W., Taylor, C.A.: Improving geometric model construction for blood flow modeling. IEEE Eng. Med. Biol. Mag. **18** (1999) 33–39
6. Westin, C.F., Lorigo, L.M., Faugeras, O., Grimson, W.E.L., Dawson, S., Norbash, A., Kinkinis, R.: Segmentation by adaptive geodesic active contours. In: MICCAI. (2000) 266–275
7. McInerney, T., Terzopoulos, D.: Medical image segmentation using topologically adaptable snakes. In: CVRMed. Volume 905 of LNCS., Springer-Verlag Berlin (1995) 92–101
8. Masutani, Y., Schiemann, T., Hohne, K.H.: Vascular shape segmentation and structure extraction using a shape-based region-growing model. In: MICCAI. (1998) 1242–1249
9. Parker, D.L., Chapman, E., Roberts, J.A., Alexander, A.L., Tsuruda, J.S.: Enhanced image detail using continuity in the MIP Z-buffer: Applications to magnetic resonance angiography. JMRI **11** (2000) 378–388
10. Aylward, S., Bullitt, E.: Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. IEEE Trans. Med. Imag. **21** (2002) 61–75
11. Krissian, K., Maladain, G., Vaillant, R., Trousset, Y., Ayache, N.: Model-based multiscale detection of 3D vessels. In: CVPR. (1998) 722–727
12. Sato, Y., Nakajima, S., Shiraga, N., Atsumi, H., Yoshida, S., Koller, T., Gerig, G., Kikinis, R.: 3D multi-scale line filter for segmentation and visialization of curvilinear structures in medical images. MedIA **2** (1998) 143–168
13. Fung, Y.C.: Biomechanics: Circulation. Second edn. Springer-Verlag (1996)
14. Barash, D.: A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. PAMI **24** (2002) 844–847
15. Weickert, J.: Coherence-enhancing diffusion filtering. IJCV **31** (1999) 111–127
16. Mignotte, M., Collet, C., Perez, P., Bouthemy, P.: Sonar image segmentation using an unsupervised hierarchical MRF model. IEEE Trans. Image Process. **9** (2000) 1216–1231
17. Cui, Y.T., Huang, Q.: Character extraction of license plates from video. In: CVPR. (1997) 502–507
18. Li, S.Z.: Markov random field modeling in image analysis. Springer-Verlag Tokyo (2001)
19. Besag, J.: Spatial interaction and the statistical analysis of lattice systems (with discussion). JRSS, Series B (Methodological) **36** (1974) 192–236
20. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: ICCV. (1998) 839–846
21. Knutsson, H.: Representing local structure using tensors. In: The 6th Scandinavian Conf. on Image Analysis. (1989) 244–251
22. Granlund, G., Knutssan, H.: Signal processing for computer vision. Kluwer Academic Publishers (1995)
23. Besag, J.: On the statistical analysis of dirty pictures. JRSS, Series B (Methodological) **48** (1986) 259–302
24. Leemput, K.V., Maes, F., Vandermeulen, D., Colchester, A., Suetens, P.: Automated segmentation of MS lesions from multi-channel MR images. In: MICCAI. (1999) 11–21

# Weighted Minimal Hypersurfaces and Their Applications in Computer Vision

Bastian Goldlücke and Marcus Magnor

Graphics - Optics - Vision
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
{bg,magnor}@mpii.de

**Abstract.** Many interesting problems in computer vision can be formulated as a minimization problem for an *energy functional*. If this functional is given as an integral of a scalar-valued weight function over an unknown hypersurface, then the minimal surface we are looking for can be determined as a solution of the functional's Euler-Lagrange equation. This paper deals with a general class of weight functions that may depend on the surface point and normal. By making use of a mathematical tool called *the method of the moving frame*, we are able to derive the Euler-Lagrange equation in arbitrary-dimensional space and without the need for any surface parameterization. Our work generalizes existing proofs, and we demonstrate that it yields the correct evolution equations for a variety of previous computer vision techniques which can be expressed in terms of our theoretical framework. In addition, problems involving minimal hypersurfaces in dimensions higher than three, which were previously impossible to solve in practice, can now be introduced and handled by generalized versions of existing algorithms. As one example, we sketch a novel idea how to reconstruct temporally coherent geometry from multiple video streams.

## 1 Introduction

A popular and successful way to treat many computer vision problems is to formulate their solution as a hypersurface which minimizes an energy functional given by a weighted area integral. In this paper, we want to expose, generalize and solve the mathematical problem which lies at the very heart of all of these methods. The aim is to find a $k$-dimensional regular hypersurface $\Sigma \subset \mathbb{R}^n$ which minimizes the energy functional

$$\mathcal{A}(\Sigma) := \int_\Sigma \Phi \, dA. \tag{1}$$

We will only investigate the case of codimension one, so throughout this text, $k = n - 1$. Such a surface is called a *weighted minimal hypersurface* with respect to the weight function. This function shall be as general as required in practice, so we allow it to depend on the surface point $s$ as well as the surface normal $\mathbf{n}$.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 366–378, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this paper, we derive a very elegant and short proof of the necessary minimality condition:

**Theorem.** A $k$-dimensional surface $\Sigma \subset \mathbb{R}^{k+1}$ which minimizes the functional $\mathcal{A}(\Sigma) := \int_\Sigma \Phi(s, \mathbf{n}(s))\, dA(s)$ satisfies the Euler-Lagrange equation

$$\langle \Phi_s, \mathbf{n} \rangle - \mathrm{Tr}(\mathbf{S})\, \Phi + \mathrm{div}_\Sigma(\Phi_\mathbf{n}) = 0, \tag{2}$$

where $\mathbf{S}$ is the shape operator of the surface, also known as the Weingarten map or second fundamental tensor. Using standard techniques, a local minimum can be obtained as a stationary solution to a corresponding surface evolution equation. Since this surface evolution can be implemented and solved in practice, the Theorem yields a generic solution to all problems of the form (1) for practical applications. In this work, we set aside the problems of convergence and local minima, as they are far from being solved yet. See e.g. [1] for a detailed analysis. This paper has thus two main contributions:

*Unification*: We unite a very general class of problems into a common mathematical framework. This kind of minimization problem arises in numerous contexts in computer vision, with dimension $n \leq 3$ and various choices of $\Phi$. A few select examples are summarized in Sect. 5, among them the method of geodesic snakes for segmentation as well as a very general multi-view 3D reconstruction technique. Our theorem yields the correct surface evolution equations for all of them.

*Generalization*: Our result is valid in arbitrary dimension. We are not aware of a previously existing treatment in computer vision literature of this generality. Until now, the theorem has been proved separately in dimensions $k = 1$ and $k = 2$, using local coordinates on the surface [2]. The now freely selectable number of surface dimensions opens up new possibilities for future applications. As one example, we generalize the static 3D reconstruction of a surface towards a space-time reconstruction of an evolving surface, which can be viewed as a 3D volume in 4D space. The proposed method treats all frames of multiple video sequences simultaneously in order to provide a temporally coherent result.

In the special case that $\Phi = 1$ is constant, the problem of minimizing (1) is reduced to finding a standard minimal surface, which is defined to locally minimize area. As we deal with a generalization, it seems reasonable to adopt the same mathematical tools used in that context [3]. A brief review of this framework, known as the *method of the moving frame*, is given in Sect. 2. However, we are forced to assume that the reader has at least some familiarity with differential geometry, preferably of frame bundles. For improved readability, we organized the paper in such a way that Sect. 2 as well as Sect. 3, in which we prove our Theorem (2), can be skipped. The transition from the Euler-Lagrange equation to a surface and further to a level set evolution equation is reviewed in Sect. 4, where we also discuss some necessary implementation details. Applications are introduced in the last two sections. In Sect. 5, we summarize a few existing computer vision techniques in order to demonstrate how they fit into our framework. A novel idea for space-time consistent reconstruction is presented in Sect. 6.

## 2 Mathematical Framework

Our goal is to give a general proof that surfaces minimizing (1) can be obtained as a solution of the Euler-Lagrange equation (2) for the energy functional. The mathematical tool of choice is called the *method of the moving frame*. This section is intended to give a brief overview of this framework.

Any minimal surface $\Sigma$ of the functional $\mathcal{A}$ is a *critical point* of the functional, i.e., in first order the value of the functional does not change under a small variation of the surface. This restriction is known as the functional's *Euler-Lagrange equation*. We are now going to give a, necessarily brief, overview of the mathematical framework in which this equation can be derived. For an excellent and thorough introduction, the reader is referred to [3].

We have to investigate how the functional behaves with respect to first order variations of the surface. To this end, let

$$X : \Sigma \times (-\epsilon, \epsilon) \to \mathbb{R}^n$$

be a variation of $\Sigma$ with compact support, then for each $\tau \in (-\epsilon, \epsilon)$ a regular surface $\Sigma_\tau \in \mathbb{R}^n$ is given by $X(\Sigma, \tau)$. For each $(s, \tau) \in \Sigma \times (-\epsilon, \epsilon)$, let

$$\{\mathbf{e}_1(s,\tau), \dots, \mathbf{e}_n(s,\tau) =: \mathbf{n}(s,\tau)\}$$

be an orthonormal frame for the surface $\Sigma_\tau$ at $s$ with $\mathbf{e}_n = \mathbf{n}$ normal to the tangent plane $T_s\Sigma_\tau$. The restrictions $\omega^i$ of the Maurer-Cartan forms of $\mathbb{R}^n$ to this frame are defined by

$$dX \;=\; \mathbf{e}_i\, \omega^i. \tag{3}$$

Throughout this text we use the Einstein convention for sums, which means that we implicitly compute the sum from 1 to $n$ over all indices appearing twice on the same side of an equation. Because the frame is adapted to $\Sigma_\tau$ in the above sense, the forms $\omega^1$ to $\omega^k$ are its usual dual forms on the surface. The *connection 1-forms* $\omega_i^j$ are defined by

$$d\mathbf{e}_i \;=\; \mathbf{e}_j\, \omega_i^j \tag{4}$$

and satisfy the *structure equations*

$$d\omega^i \;=\; -\omega_j^i \wedge \omega^j \qquad d\omega_j^i \;=\; \omega_k^i \wedge \omega_j^k, \tag{5}$$

which can be deduced by differentiating the definitions.

From the connection forms stems the true power of this method. They allow us to express derivatives of the frame, in particular of the normal, in terms of objects which are *part of the frame bundle themselves*. This is the one reason why we will never need local coordinates, because all necessary information about the embedding of the surface in space is encoded in the connection forms.

From the Euclidean structure on $\mathbb{R}^n$ it follows that the connection 1-forms are skew-symmetric, $\omega_i^j = -\omega_j^i$. The connection forms $\omega_i^n$ can be expressed in

the base $\{\omega^1, \ldots, \omega^k, d\tau\}$, courtesy of Cartan's Lemma [4]. To see this, first note that because of definition (3)

$$\omega^n \;=\; \langle dX, \mathbf{n}\rangle \;=\; \frac{\partial X}{\partial \tau}\, d\tau \;=:\; f\, d\tau. \tag{6}$$

Differentiating this equation yields together with (5)

$$df \wedge d\tau \;+\; \sum_{i=1}^{k} \omega_i^n \wedge \omega^i \;=\; 0,$$

therefore, by Cartan's Lemma, there exist functions $h_{ij}$ such that

$$\begin{bmatrix} \omega_1^n \\ \vdots \\ \omega_k^n \\ df \end{bmatrix} \;=\; \begin{bmatrix} h_{11} & \ldots & h_{1k} & f_1 \\ \vdots & \ddots & \vdots & \vdots \\ h_{k1} & \ldots & h_{kk} & f_k \\ f_1 & \ldots & f_k & f_n \end{bmatrix} \begin{bmatrix} \omega^1 \\ \vdots \\ \omega^k \\ d\tau \end{bmatrix}. \tag{7}$$

The top-left part $\mathbf{S} := (h_{ij})$ of this matrix is called the *shape operator*, and is closely related to the curvature of $\Sigma_\tau$. In the lower dimensional cases, its entries are commonly known as follows:

- If $k = 1$, i.e. $\Sigma_\tau$ is a curve in $\mathbb{R}^2$, the sole coefficient $h_{11}$ equals the scalar valued curvature usually denoted by $\kappa$.
- If on the other hand $k = 2$, i.e. $\Sigma$ is a regular surface in $\mathbb{R}^3$, the entries of $\mathbf{S}$ are the coefficients of the second fundamental form of $\Sigma_\tau$. More precisely,

$$II \;=\; \begin{bmatrix}\omega^1 & \omega^2\end{bmatrix} \mathbf{S} \begin{bmatrix}\omega^1 \\ \omega^2\end{bmatrix} \;=\; h_{11}(\omega^1)^2 + 2h_{12}\omega^1\omega^2 + h_{22}(\omega^2)^2.$$

  Thus $H = \frac{1}{k}\mathrm{Tr}\,(\mathbf{S}) = \frac{1}{k}\sum_{i=1}^{k} h_{ii}$ is the mean curvature of the surface.

The $f_i$ are just the directional derivatives of $f$ in the directions of the $\mathbf{e}_i$. Using the structure equations (5), we immediately deduce an important relation for the area form $dA$ on $\Sigma_\tau$:

$$dA =: \omega_A = \omega^1 \wedge \ldots \wedge \omega^k \implies d\omega_A \;=\; -\mathrm{Tr}\,(\mathbf{S})\; \omega_A \wedge \omega^n, \tag{8}$$

We introduce the notation $\omega_A$ to remind the reader of the fact that the area element $dA$ indeed is a differential form of degree $k$. Note that area in our sense does not imply "two-dimensional".

Finally, we need a notion of an 'integration by parts' for a surface integral. First, we generalize the usual operators from vector analysis to vector fields $\mathbf{v}$ and functions $f$ on $\Sigma$:

$$\mathrm{div}_\Sigma(\mathbf{v}) \;:=\; \sum_{i=1}^{k} \frac{\partial v^i}{\partial \mathbf{e}_i} \text{ with the expansion } \mathbf{v} \;=\; v^i\, \mathbf{e}_i, \text{ and}$$

$$\nabla_\Sigma f \;:=\; \sum_{i=1}^{k} \frac{\partial f}{\partial \mathbf{e}_i}\, \mathbf{e}_i \;=\; \sum_{i=1}^{k} f_i e_i.$$

Using the definitions and the product rule, we derive a generalization of an identity well-known from classical vector analysis,

$$\operatorname{div}_\Sigma(\mathbf{v}f) \;=\; \langle \mathbf{v}, \nabla_\Sigma f\rangle \;+\; \operatorname{div}_\Sigma(\mathbf{v})\, f, \tag{9}$$

which will be useful later as one possibility of shifting partial derivatives from one object to another. A second possibility is given by Gauss' Theorem for surfaces, which in our context reads

$$\int_\Sigma \operatorname{div}_\Sigma(\mathbf{v})\, dA \;=\; -\int_\Sigma \operatorname{Tr}(\mathbf{S})\, \langle \mathbf{v}, \mathbf{n}\rangle\, dA. \tag{10}$$

Note that $\mathbf{v}$ does not have to be tangential to $\Sigma$. Since we assume that all our surfaces are closed, the boundary term usually contributing to the formula has vanished.

We have now collected all the necessary tools to derive the Euler-Lagrange equation of $\mathcal{A}$, and do so in the next section. In Sect. 4, this will yield an evolution equation for the level sets of a function on $\mathbb{R}^n$.

## 3 Euler-Lagrange Equation

In this section we employ the mathematical framework to derive the Euler-Lagrange equation of the functional $\mathcal{A}$. The arguments can be followed just by abstract manipulation of symbols, without the need to understand all of the reasons which lead to the governing rules presented in Sect. 2.

The desired equation characterizes critical points of $\mathcal{A}$, and is given by the derivation of the functional with respect to $\tau$ at $\tau = 0$. We assume that $\Phi = \Phi(s, \mathbf{n})$ is a function of the surface point $s$ and the normal $\mathbf{n}(s)$ at this point. Since $\Phi$ maps from $\mathbb{R}^n \times \mathbb{S}^n$, $\Phi_\mathbf{n}(s, \mathbf{n})$ is tangent to the unit sphere of $\mathbb{R}^n$ at $\mathbf{n}$, so we have the important relation

$$\langle \Phi_\mathbf{n}(s, \mathbf{n}), \mathbf{n}\rangle \;=\; 0. \tag{11}$$

This fact was overlooked in previous publications, which is the reason why our final equation is considerably simpler.

Let us now turn to the computation of the Euler-Lagrange equation. Using the Lie-derivative

$$\mathcal{L}_\mathbf{v}\omega \;=\; \mathbf{v} \rightharpoonup d\omega \;+\; d(\mathbf{v} \rightharpoonup \omega) \tag{12}$$

of a differential form $\omega$ in the direction of $\mathbf{v}$, we obtain

$$\begin{aligned}
\left.\frac{d}{d\tau}\right|_{\tau=0} \mathcal{A}(\Sigma_\tau) \;&\overset{(a)}{=}\; \int_\Sigma \mathcal{L}_{\frac{\partial}{\partial\tau}}(\Phi\, \omega_A) \;\overset{(b)}{=}\; \int_\Sigma \frac{\partial}{\partial\tau} \rightharpoonup d(\Phi\, \omega_A) \\
&\overset{(c)}{=} \int_\Sigma \frac{\partial}{\partial\tau} \rightharpoonup (d\Phi \wedge \omega_A \;+\; \Phi\, d\omega_A) \\
&\overset{(d)}{=} \int_\Sigma \frac{\partial}{\partial\tau} \rightharpoonup \left(\langle \Phi_s, \mathbf{e}_i\rangle\, \omega^i \wedge \omega_A \;+\; \Phi_\mathbf{n}\, d\mathbf{n} \wedge \omega_A \;-\; \operatorname{Tr}(\mathbf{S})\, \Phi\, \omega_A \wedge \omega^n\right) \\
&\overset{(e)}{=} \int_\Sigma \left[\left(\langle \Phi_s, \mathbf{n}\rangle - \operatorname{Tr}(\mathbf{S})\, \Phi\right) f\, \omega_A \;+\; \frac{\partial}{\partial\tau} \rightharpoonup (\Phi_\mathbf{n}\, d\mathbf{n} \wedge \omega_A)\right].
\end{aligned} \tag{13}$$

The five equalities above are justified by the following arguments:

a. A generalization of the 'Differentiation under the integral'-rule in classic calculus [3].
b. Cartan's rule (12) for expressing the Lie derivative and using the fact that $\omega^1(\mathbf{n}) = \cdots = \omega^k(\mathbf{n}) = 0$. Note that $\frac{\partial}{\partial\tau}$ is parallel to $\mathbf{n}$, so this equation also holds for $\frac{\partial}{\partial\tau}$.
c. Product rule for differential forms, note that $\Phi$ is a 0-form.
d. Expansion of $d\Phi = \Phi_s\, dX + \Phi_{\mathbf{n}}\, d\mathbf{n} = \langle \Phi_s, \mathbf{e}_i \rangle\, \omega^i + \Phi_{\mathbf{n}}\, d\mathbf{n}$. Here we inserted the definition (3) of the restrictions $\omega^i$. The last term is due to (8).
e. Linearity of the left hook and again $\omega^1(\mathbf{n}) = \cdots = \omega^k(\mathbf{n}) = 0$. From (6), it follows that $\omega^n(\frac{\partial}{\partial\tau}) = f d\tau(\frac{\partial}{\partial\tau}) = f$.

We now turn our attention to the second term of the last integral. Inserting the definition (4) of the connection 1-forms and afterwards the expansion of the connection forms (7) due to Cartan's Lemma, we get

$$\begin{aligned} &\frac{\partial}{\partial\tau} \rightharpoonup (\Phi_{\mathbf{n}}\, d\mathbf{n} \wedge \omega_A) \;=\; \frac{\partial}{\partial\tau} \rightharpoonup \left(\langle \Phi_{\mathbf{n}}, \mathbf{e}_j \rangle\; \omega_n^j \wedge \omega_A\right) \\ &= \frac{\partial}{\partial\tau} \rightharpoonup (-\langle \Phi_{\mathbf{n}}, \nabla_\Sigma f \rangle\; d\tau \wedge \omega_A) \;=\; -\langle \Phi_{\mathbf{n}}, \nabla_\Sigma f \rangle\; \omega_A \\ &= \operatorname{div}_\Sigma(\Phi_{\mathbf{n}})\, f\, \omega_A \;-\; \operatorname{div}_\Sigma(\Phi_{\mathbf{n}}\, f)\; \omega_A. \end{aligned} \tag{14}$$

In the last equality, we have shifted derivatives using the product rule (9). We can finally compute the integral over the left term using Gauss' Theorem (10):

$$\int_\Sigma -\operatorname{div}_\Sigma(\Phi_{\mathbf{n}}\, f)\; dA \;=\; \int_\Sigma \operatorname{Tr}(\mathbf{S})\, \langle \Phi_{\mathbf{n}}, \mathbf{n} \rangle\; f\, dA \;=\; 0.$$

It vanishes due to (11). When we thus put equations (13) and (14) together, we see that we have derived

$$\left.\frac{d}{d\tau}\right|_{\tau=0} \mathcal{A}(\Sigma_\tau) = \int_\Sigma \left(\langle \Phi_s, \mathbf{n} \rangle - \operatorname{Tr}(\mathbf{S})\, \Phi + \operatorname{div}_\Sigma(\Phi_{\mathbf{n}})\right)\, f\, dA.$$

Since for a critical point this expression must be zero for any variation and hence for any $f$, we have arrived at the Euler-Lagrange equation of the functional

$$\langle \Phi_s, \mathbf{n} \rangle \;-\; \operatorname{Tr}(\mathbf{S})\, \Phi \;+\; \operatorname{div}_\Sigma(\Phi_{\mathbf{n}}) \;=\; 0, \tag{15}$$

and thus proved our Theorem (2).

## 4 Corresponding Level Set Equation

Level sets represent an efficient way to implement a surface evolution [5,6], and are by now a well-established technique with a wide area of applications [7]. We will briefly review the transition from (15) to a surface evolution equation followed by one for a level set in this section. For the remainder of the text, let

$$\Psi := \langle \Phi_s, \mathbf{n} \rangle \;-\; \operatorname{Tr}(\mathbf{S})\, \Phi \;+\; \operatorname{div}_\Sigma(\Phi_{\mathbf{n}}).$$

A surface $\hat{\Sigma}$ which is a solution to the Euler-Lagrange equation $\Psi = 0$ is likewise a stationary solution to a surface evolution equation, where $\Psi$ describes a force in the normal direction:

$$\frac{\partial}{\partial \tau} \Sigma_\tau \;=\; \Psi \mathbf{n}. \tag{16}$$

If we start with an initial surface $\Sigma_0$ and let the surface evolve using this equation, it will eventually converge to a local minimum of $\mathcal{A}$. Instead of implementing a surface evolution directly, we can make use of the level set idea. We express the surfaces $\Sigma_\tau$ for each parameter value $\tau \geq 0$ as the zero level sets of a regular function $u : \mathbb{R}^n \times \mathbb{R}^{\geq 0} \to \mathbb{R},\; u(s, \tau) = 0 \;\Leftrightarrow\; s \in \Sigma_\tau$.

We require $u(\cdot, \tau)$ to be positive in the volume enclosed by $\Sigma_\tau$, thus if $\nabla$ is the gradient operator for the spatial coordinates of $u$, we can compute the outer normal using

$$\mathbf{n} \;=\; -\frac{\nabla u}{|\nabla u|} \quad\Longrightarrow\quad |\nabla u| \;=\; -\left\langle \nabla u, \mathbf{n} \right\rangle .$$

Taking the derivative of $u(s, \tau) = 0$ with respect to $\tau$ and inserting (16), we deduce the evolution equation for $u$ to be

$$\frac{\partial}{\partial \tau} u \;=\; -\left\langle \nabla u, \frac{\partial}{\partial \tau} \Sigma_\tau \right\rangle \;=\; -\left\langle \nabla u, \mathbf{n} \right\rangle \Psi \;=\; \Psi \, |\nabla u| . \tag{17}$$

Using the identity

$$\mathrm{Tr}\,(\mathbf{S}) \;=\; \mathrm{div}\left( \frac{\nabla u}{|\nabla u|} \right)$$

for the curvature of the level sets of $u$ and the definition of $\Psi$, we arrive at the final reformulation of (16) in terms of a level set evolution:

$$\frac{\partial}{\partial \tau} u \;=\; \left[ -\,\mathrm{div}\left( \Phi \cdot \frac{\nabla u}{|\nabla u|} \right) \;+\; \mathrm{div}_\Sigma(\Phi_{\mathbf{n}}) \right] |\nabla u| .$$

Note that the derivatives of $\Phi$ can be computed numerically. Thus, it is not necessary to compute an explicit expression for them manually, which would be very cumbersome for more difficult functionals. Instead, in an existing implementation of the evolution for a general function $\Phi$, essentially any functional can be plugged in.

## 5 Known Applications in Computer Vision

Among the first variational methods which were successfully utilized for computer vision problems was the one now widely known as *Geodesic Active Contours* [8]. While originally designed for segmentation in 2D, it quickly became clear that it could be generalized to 3D [9], and also applied to other tasks. It is particularly attractive for modeling surfaces from point clouds [10,11]. Geodesic contours were also employed for 2D detection and tracking of moving objects [12].

Also well analyzed in theory is how to employ minimal surfaces for 3D reconstruction of static objects from multiple views [13]. This technique was recently extended to simultaneously estimate the radiance of surfaces, and demonstrated to give good results in practice [14].

We will briefly review the above methods to demonstrate that all of them fit into our framework. In particular, our theorem applies to all of them and yields the correct surface evolution equations.

### 5.1 Segmentation via Geodesic Active Contours

Caselles, Kimmel and Sapiro realized that the energy which is minimized in the classical *snakes* approach [15] can be rewritten in terms of a geodesic computation in a Riemannian space by means of Maupertuis' Principle. The goal is to compute a contour curve $\mathcal{C}$ in an image $I$ which is attracted by edges in the image while remaining reasonably smooth. Their final energy functional took the form

$$\mathcal{A}(\mathcal{C}) := \int_{\mathcal{C}} g \circ |\nabla I| \, ds,$$
$$\text{where } g : \mathbb{R}^+ \to \mathbb{R}^+ \text{ is strictly decreasing with } \lim_{r \to 0} g(r) = 0.$$

$\nabla I$ acts as an edge detector, while $g$ controls how image gradients are interpreted as energies. The main purpose of $g$ is to act as a *stopping function*: The flow of the curve should cease when it arrives at object boundaries. Because the integral is minimized, the contour will move towards regions of high gradient. The smoothness requirement is enforced by the curvature term in equation (2). Note that $g \circ |\nabla I|$ depends only on the surface point and not on the normal, so the rightmost term in the Euler-Lagrange equation vanishes.

Essentially the same functional can be applied to 3D segmentation [9], where the source image $I$ is replaced by a volumetric set of data, and the unknown curve $\mathcal{C}$ by an unknown 2D surface. Based on another derivation of the conformal length minimizing flow by Kichenassamy et al. [16], Zhao et al. [11,17] chose an Euclidean distance function instead of an edge-based stopping potential for $\Phi$ to model surfaces from unstructured data sets.

### 5.2 Tracking

Paragios and Deriche combine geodesic active contours and a motion detection term in a single energy functional to track moving objects in a sequence of images [12]:

$$\mathcal{A}(\mathcal{C}) := \int_{\mathcal{C}} \gamma \underbrace{G_{\sigma_D} \circ I_D}_{Motion} + (1-\gamma) \underbrace{G_{\sigma_T} \circ |\nabla I|}_{Contours} ds,$$

where $G_\sigma$ is a Gaussian with variance $\sigma$. The user-defined parameter $\gamma$ weights the influence of the motion detection term against the boundary localization. The Gaussians play the same role as $g$ in geodesic contours, their variances

$\sigma_T$ and $\sigma_D$ are derived from the image statistics. The image $I_D$ is designed to detect boundaries of moving regions in the current image $I$ of the sequence, and constructed using a Bayesian model which takes into account the pixel differences to the previous frame.

### 5.3 3D Reconstruction

As a first step, Faugeras and Keriven [13] give a simple functional in dimension $n = 3$ for static 3D scene reconstruction which does not depend on the surface normal. It can be viewed as a space-carving approach [18] generalized from discrete voxels to a continuous surface model.

Let $C_1, \dots, C_l$ be a number of cameras which project a scene in $\mathbb{R}^3$ onto images $I_k$ via projections $\pi_k : \mathbb{R}^3 \to \mathbb{R}^2$. For each point $s \in \mathbb{R}^3$, let $\nu_k(s)$ denote whether or not $s$ is visible in camera $k$ in the presence of a surface $\Sigma$. $\nu_k(s)$ is defined to be one if $s$ is visible, and zero otherwise. A measure of how good a surface $\Sigma$ as a model of the scene geometry really is in accordance with a given set of images can be obtained as follows: Each surface point is projected into the set of images where it is visible, and the differences between the pixel colors for each pair of images are computed and summed up to get an error measure for the surface point. This error is integrated over the surface to get the total error. In mathematical notation,

$$\begin{aligned} \mathcal{A}(\Sigma) &:= \int_\Sigma \Phi^S dA, \text{ where} \\ \Phi^S(s) &:= \frac{1}{V_s(V_s-1)} \sum_{i,j=1}^{l} \nu_i(s)\nu_j(s) \cdot \left\| I_i \circ \pi_i(s) - I_j \circ \pi_j(s) \right\|_\infty . \end{aligned}$$

The number $V_s$ of cameras able to see a point $s$ is used to normalize the function.

Clearly, the above model is too simple to be of much use in multi-view reconstruction, since only single pixels with no regard to their neighborhoods are compared. A better functional was therefore suggested by Faugeras and Keriven, and can be applied using the results on how the evolution depends on the current normals. We present a slight modification of their original approach here. Our functional only depends on invariant surface properties and does not make use of geometric objects in the source camera views.

To each surface point $s$, we associate a small rectangle $\square_{s,\mathbf{n}}$ in the tangent plane $T_s\Sigma$. In order to invariantly determine its orientation within the plane, we align the sides with the principal curvature directions. This rectangle is then projected into the images, and the normalized cross-correlation over the projected areas is computed. We choose the length of the rectangle sides to be inversely proportional to the curvature in the corresponding direction, up to a certain maximum, because the first order approximation of the surface by its tangent plane is valid over a larger region if the curvature is low. The corresponding functional can be written as

$$\begin{aligned}
\mathcal{A}(\Sigma) &:= \int_{\Sigma} \Phi^C dA, \text{ where} \\
\Phi^C(s,\mathbf{n}) &:= -\frac{1}{V_s(V_s-1)} \sum_{i,j=1}^{l} \nu_i(s)\nu_j(s) \cdot \chi_{i,j}(s,\mathbf{n}) \text{ and} \\
\chi_{i,j}(s,\mathbf{n}) &:= \frac{1}{A(\square_{s,\mathbf{n}})} \int_{\square_{s,\mathbf{n}}} \left(I_i \circ \pi_i - \overline{I}_i^{s,\mathbf{n}}\right) \cdot \left(I_j \circ \pi_j - \overline{I}_j^{s,\mathbf{n}}\right) dA.
\end{aligned}$$

The correlation integral has to be normalized using the area $A(\square_{s,\mathbf{n}})$ of the square. The mean values are computed using

$$\overline{I}_i^{s,\mathbf{n}} := \frac{1}{A(\square_{s,\mathbf{n}})} \int_{\square_{s,\mathbf{n}}} I_i \circ \pi_i \, dA.$$

When this functional is minimized, not only the position, but also the surface normal is adjusted to best match the images. This approach can also be employed to improve the normals for a known geometry approximation, i.e., the visual hull. When a segmentation of the images into background and foreground objects can be obtained, the visual hull also constitutes a good initial surface $\Sigma_0$ for the evolution equation (16), since it is by construction a conservative estimate of the object regions.

### 5.4 Reflectance Estimation

Jin, Soatto and Yezzi combine the reconstruction framework with a simultaneous reflectance estimation [14]. They use the functional

$$\mathcal{A}(\Sigma) := \int_{\Sigma} \left\| \tilde{R} - R \right\|_F^2 dA,$$

where the Frobenius norm $\|\cdot\|_F$ is employed to compute the difference of the measured radiance tensor field $\tilde{R}$ to an idealized $R$ obtained from a reflection model, which depends on the surface $\Sigma$.

As claimed previously, all of the problems reviewed in this section are of the form required by the main theorem, and can thus be subsumed under the unifying framework presented in this paper.

## 6 A Novel Technique: Space-Time 3D Reconstruction

In this section and for the first time, we are going to exploit the fact that we can now handle variational problems posed in higher-dimensional space. We present only one of many new applications one can think of. For instance, the additional degrees of freedom could also be used to optimize parameters defined in each surface point. We employ them to introduce a temporal dimension, which allows us to compute temporally coherent estimates for the scene geometry reconstruction from multiple video streams.

Since our equation is valid in arbitrary dimension, we can interpret it in dimension $n = 4$ as an evolution equation for a hypersurface in space-time. Its cross sections with planes of constant time $\{t = t_0\}$ yield the scene geometry at the time instant $t_0$. That way, a global optimum for the scene geometry, including the normals, *and its change over time* can be found which takes into account *every frame at every time instant* simultaneously. Moreover, the minimization problem is formulated and can be solved in an elegant mathematical context.

In order to distinguish normal surfaces in $\mathbb{R}^3$ from hypersurfaces, we will denote the latter by $\mathfrak{H}$. Points in space-time $\mathbb{R}^4$ are written as $x = (s, t) \in \mathbb{R}^3 \times \mathbb{R}$. A hypersurface $\mathfrak{H}$ gives rise to a family $(\Sigma_t)$ of regular surfaces

$$\Sigma_t \;:=\; \mathfrak{H} \cap (\mathbb{R}^3, t) \;\subset\; \mathbb{R}^3$$

for each time instant $t$. We also have to deal with time-dependent visibilities $\nu^t(s)$ depending on point $s$ and the surface $\Sigma_t$ at time-instant $t$, as well as time-dependent images $I^t$ from each camera.

The resulting functional looks almost identical to the one in Sect. 5.3:

$$\begin{aligned}
\mathcal{A}(\mathfrak{H}) \;&:=\; \int_{\mathfrak{H}} \Phi^G dV, \text{ where} \\
\Phi^G(x, \mathbf{n}) \;&:=\; -\frac{1}{V_{s,t}(V_{s,t} - 1)} \sum_{i,j=1}^{l} \nu_i^t(s)\nu_j^t(s) \cdot \chi_{i,j}^t(s, \mathbf{n}^t). \\
\chi_{i,j}^t(s, \mathbf{n}^t) \;&:=\; \frac{1}{A(\square_{s,\mathbf{n}^t})} \int_{\square_{s,\mathbf{n}^t}} \left(I_i^t \circ \pi_i - \overline{I}_i^{x,\mathbf{n}}\right) \cdot \left(I_j \circ \pi_j - \overline{I}_j^{x,\mathbf{n}}\right) \, dA.
\end{aligned}$$

The normal $\mathbf{n}^t$ to the surface $\Sigma_t$ is the projection of the normal to the hypersurface $\mathfrak{H}$ onto the tangent space of $\Sigma_t$, which is a subspace of the tangent space of $\mathfrak{H}$. Note that the square $\square_{s,\mathbf{n}^t}$ lies inside the tangent plane of $\Sigma_t$. The mean values $\overline{I}_i^{x,\mathbf{n}}$ are of course also computed using the images at time $t$.

When this functional is minimized, two constraints are optimized simultaneously. First, each surface $\Sigma_t$ together with its normals is selected to best match the images at that time instant. Second, a smooth change of the surfaces $\Sigma_t$ with time is encouraged because of the curvature term in the Euler-Lagrange equation. Our experiments with real-world data using a parallel implementation of this scheme gave very promising results, which we are going to present in a future publication.

## 7 Conclusion

Using the mathematical tool of the *method of the moving frame*, we have derived the Euler-Lagrange equations for weighted minimal surfaces in arbitrary dimensions. We allowed for weight functions general enough to cover the variational problems encountered in computer vision research. Previously, existing proofs used local coordinates and were restricted to dimensions two or three, so our approach is more general. As demonstrated by several examples, weighted minimal

surfaces lie at the heart of several well-established computer vision techniques. Our result for arbitrarily high dimensions paves the way for new, future research. In particular, we sketched a technique designed to achieve temporal coherence in 3D reconstruction from multiple video streams. In the near future, we are going to experimentally investigate its possibilities.

## References


1. Chen, Y., Giga, Y., Goto, S.: Uniqueness and existence of viscosity solutions of generalized mean curvature flow. Journal of Differential Geometry **33** (1991) 749–786
2. Siddiqi, K., Lauziere, Y.B., Tannenbaum, A., Zucker, S.W.: Area and length minimizing flows for shape segmentation. IEEE Transactions on Image Processing **3** (1998) 433–443
3. Clelland, J.: MSRI Workshop on Lie groups and the method of moving frames. Lecture Notes. Department of Mathematics, University of Colorado (1999) *http://spot.Colorado.EDU/∼jnc/MSRI.html*.
4. Sharpe, R.: Differential Geometry. Graduate Texts in Mathematics. Springer (1997)
5. Osher, S., Sethian, J.: Fronts propagating with curvature dependent speed: Algorithms based on the Hamilton-Jacobi formulation. Journal of Computational Physics **79** (1988) 12–49
6. Chop, D.: Computing minimal surfaces via level set curvature flow. Journal of Computational Physics **106** (1993) 77–91
7. Sethian, J.A.: Level Set Methods and Fast Marching Methods. 2nd edn. Monographs on Applied and Computational Mathematics. Cambridge University Press (1999)
8. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. International Conference on Computer Vision. (1995) 694–699
9. Caselles, V., Kimmel, R., Sapiro, G., Sbert, C.: Three dimensional object modeling via minimal surfaces. In: Proc. European Conference on Computer Vision. Volume 1., Springer (1996) 97–106
10. Caselles, V., Kimmel, R., Sapiro, G., Sbert, C.: Minimal surfaces based object segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997) 394–398
11. Zhao, H., Osher, S., Fedkiw, R.: Fast surface reconstruction using the level set method. 1st IEEE Workshop on Variational and Level Set Methods, 8th ICCV **80** (2001) 194–202
12. Paragios, N., Deriche, R.: Geodesic active contours and level sets for the detection and tracking of moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 266–280
13. Faugeras, O., Keriven, R.: Variational principles, surface evolution, PDE's, level set methods and the stereo problem. IEEE Transactions on Image Processing **3** (1998) 336–344
14. Jin, H., Soatto, S., Yezzi, A.: Multi-view stereo beyond Lambert. In: IEEE Conference on Computer Vision and Pattern Recognition. Volume I., Madison, Wisconsin, USA (2003) 171–178
15. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. International Journal of Computer Vision **1** (1988) 321–331

16. Kichenassamy, S., Kumar, A., Olver, P.J., Tannenbaum, A., Yezzi, A.: Gradient flows and geometric active contour models. In: ICCV. (1995) 810–815
17. Zhao, H., Osher, S., Merriman, B., Kang, M.: Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. In: Computer Vision and Image Understanding. (2000) 295–319
18. Kutukalos, K.N., Seitz, S.M.: A theory of shape by space carving. International Journal of Computer Vision **38** (2000) 197–216

# Interpolating Novel Views from Image Sequences by Probabilistic Depth Carving

Annie Yao and Andrew Calway

Department of Computer Science
University of Bristol, UK
{yao,andrew}@cs.bris.ac.uk

**Abstract.** We describe a novel approach to view interpolation from image sequences based on probabilistic depth carving. This builds a multivalued representation of depth for novel views consisting of likelihoods of depth samples corresponding to either opaque or free space points. The likelihoods are obtained from iterative probabilistic combination of local disparity estimates about a subset of reference frames. This avoids the difficult problem of correspondence matching across distant views and leads to an explicit representation of occlusion. Novel views are generated by combining pixel values from the reference frames based on estimates of surface points within the likelihood representation. Efficient implementation is achieved using a multiresolution framework. Results of experiments on real image sequences show that the technique is effective.

## 1 Introduction

There is great interest in developing algorithms to interpolate novel views of a static scene from a set of reference views. Applications range from web based virtual tours to video compression. Previous approaches either make direct use of 3-D representations [9,14,12,5,13,8,6] or employ 3-D information implicitly within image based transformations [15,11,2,10]. Our interest here is in techniques that use calibrated views from image sequences, usually obtained from a structure from motion algorithm, and dense correspondences to determine view centred depth maps [14,12,8]. Novel views are then generated by surface fitting and re-projecting pixel values into the virtual frames. These methods have considerable flexibility in terms of the range of views that can be interpolated and apart from visibility constraints they impose few restrictions on the positioning of the reference views. The difficulties are in obtaining robust dense depth estimates from correspondences and in deriving surface representations which result in convincing novel views. It is well known that establishing dense correspondences is problematic: wide baselines result in mismatches, whilst narrow baselines give low accuracy and increased noise sensitivity. Textureless regions also complicate the matching process. Surface fitting is also problematic, especially around depth discontinuities. Although various global optimisation techniques have been used to improve matters [12,8], the high computational costs involved make them unsuitable when dealing with a large number of reference views.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 379–390, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this paper we present a new approach to view interpolation which addresses these problems. We borrow ideas from space carving [13] and in particular its recent probabilistic formulations [6,1]. For a given viewpoint, we build a multivalued depth representation in which depth samples along a pixel viewing ray are classified as belonging either to an opaque point or to a free space point. However, unlike space carving, we build our representation not from projected pixel differences, but from disparity (depth) estimates obtained locally about reference frames. Likelihood values for depth classification are then obtained using an iterative combination of these disparity estimates based on explicit models of opacity and occlusion. In this respect we are motivated by the work of Agrawal and Davis [1] and Szeliski and Golland [17]. We call this process *probabilistic depth carving*. In essence, the local depth estimates are 'triangulated' in 3-D space in order to carve away free space points. This yields more reliable and accurate depth representations than the individual local estimates, while at the same time avoiding the problem of establishing correspondences between wide baseline views. It also provides a coherent framework within which to combine widely disparate views of a scene. Moreover, importantly for view interpolation, view centred depth representations can easily be constructed for virtual frames from those derived for the reference frames, leading to a straightforward process for generating novel views which avoids surface fitting.

The principles of depth carving were set out in [18]. In this paper we concentrate on its use for view interpolation and describe an efficient multiresolution implementation. The next two sections provide an overview of the basic ideas and the theoretical framework. The probabilistic and multiresolution implementation are then described in Sections 4-6 and Section 7 describes the view interpolation process. Examples are presented for two real image sequences.

## 2 Depth Carving

Consider the two examples illustrated in Fig. 1. In each case a novel view is being generated in frame $V$ based on local correspondence matching about three reference frames $A$, $B$ and $C$. The scene in Fig. 1a consists of a smooth surface, primarily textureless, but with a central textured region indicated by the zig-zag line. Pixel viewing rays from the reference frames are indicated as $a1$-$c1$ and $a2$-$c2$ and along each ray we have indicated in bold the range of depths having high matching values in the local correspondence analysis. Thus rays $a1$ and $c1$ have a wide range of likely depths due to the lack of texture on the surface, whilst rays $b1$ and $a2$-$c2$ have a narrow range since they intersect the textured region (we assume for simplicity that the autocorrelation function of the texture has a symmetric narrow peak about zero).

In depth carving we seek to classify sample points along viewing rays as either being opaque or in free space. This is achieved by combining the depth likelihoods obtained from the local correspondence matching using 'triangulation'. For example, in Fig. 1a, point $P$ along the novel viewing ray $v1$ is in free space, although due to the textureless surface this cannot be determined from

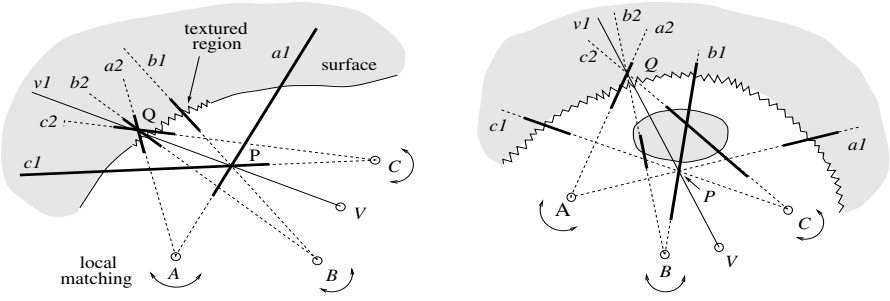



**Fig. 1.** Depth carving combines matching similarities obtained from local correspondences to obtain improved depth representations.

views $A$ and $C$; the depth is ill-defined along the rays $a1$ and $c1$. However this can be resolved using view $B$: ray $b1$ intersects the textured region and thus has well defined depth beyond point $P$, enabling the latter to be correctly classified as free space. In effect, ray $b1$ carves away the free space in front of the textured surface. This illustrates a useful property: reliable depth estimates obtained about certain reference views due to the presence of textured regions can be used to sort out ambiguous estimates elsewhere. In contrast, the classification of point $Q$ along $v1$ is straightforward since it lies within the textured region and is visible in all three views, giving a high consensus for opacity amongst rays $a2$-$c2$.

Visibility is important in depth carving. Because of possible occlusion it is not sufficient to simply combine the local depth likelihoods directly to classify points along viewing rays. This can be seen from the example in Fig. 1b, which consists of a textured background surface and a textureless foreground object. Point $Q$ in this figure is opaque and this is supported by the depth likelihoods along the ray $a2$. However the point is occluded from views $B$ and $C$ and consequently the depth likelihoods along rays $b2$ and $c2$ correspond to the occluding front surface. Thus if point $Q$ is to be correctly classified this occlusion needs to be accounted for; in effect, cancelling out the free space classification implied by rays $b2$ and $c2$. As discussed below, this can achieved by incorporating explicit models of opacity and occlusion into the carving process. The example in Fig. 1b also further illustrates the advantages of the carving process. Point $P$ lies in free space just in front of the foreground object. Rays $a1$ and $c1$ intersect textured regions on the background and hence carve away point $P$, removing the ambiguity along ray $b1$ caused by the lack of texture on the foreground object. Repeating this for other viewing rays will carve away the free space around the object. Thus, in principle and with enough reference views, the 3-D volume occupied by the object can be determined despite the lack of texture on its surface.

The key point about depth carving is therefore its ability to combine local depth likelihoods from disparate views, enabling well-defined depths to correct ambiguities in other views. Notably, since it is not based on forming correspondences across all the reference views, it also allows disparate views to be com-

bined without requiring them to contain common elements, a property often absent from other approaches. In the next section we define the explicit models of opacity and occlusion which are needed to implement depth carving.

## 3 Visible Opacity

The implementation of depth carving is based on two properties of points in a 3-D scene - opacity and visibility - and on a relationship between them which we call the visible-opacity constraint. These are defined as follows.

**Opacity**. Assuming that the scene consists only of opaque objects, then the opacity associated with a 3-D point $\mathbf{X}$ is defined as

$$\alpha(\mathbf{X}) = \begin{cases} 1 \text{ if } \mathbf{X} \text{ is an interior point} \\ 0 \text{ if } \mathbf{X} \text{ is a point in free space} \end{cases} \tag{1}$$

where an 'interior point' is a point within an object. This is a restricted form of the opacity used by Bonet and Viola [5]; unlike them we do not consider the case of transparent objects being in the scene for which $0 < \alpha(\mathbf{X}) < 1$. Note that the opacity for surface points is not defined since they occur at the transition between free space and opacity.

**Visibility**. The visibility $\upsilon(k, \mathbf{X})$ indicates whether a point $\mathbf{X}$ is visible in frame $k$, ie

$$\upsilon(k, \mathbf{X}) = \begin{cases} 1 \text{ if } \mathbf{X} \text{ is visible in frame } k \\ 0 \text{ if } \mathbf{X} \text{ is occluded from frame } k \end{cases} \tag{2}$$

where all visible points are free space points and occluded points can be interior as well as free space or surface points. Note that in general $\upsilon(k, \mathbf{X})$ need not equal $\upsilon(j, \mathbf{X})$, $k \neq j$, and also that $\upsilon(k, \mathbf{X})$ is directly related to the $\alpha(\mathbf{X})$ along the same viewing ray from frame $k$, ie

$$\upsilon(k, \mathbf{X}) = \begin{cases} 1 \text{ if } \alpha(a\mathbf{X} + (1-a)\mathbf{P}_k) = 0 \quad \forall \quad 0 < a < 1 \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

where $\mathbf{P}_k$ denotes the centre of projection (COP) for frame $k$. Thus, $\mathbf{X}$ is only visible in frame $k$ if all points preceding it along the viewing ray lie in free space and the visibility of the closest surface point is not defined - the surface boundary corresponds to the transition between visibility and occlusion.

**Visible Opacity**. Given the above two properties, given an opaque scene and the condition that *every* free space point is visible in *at least one* frame, the following relationship between opacity and visibility can be derived [18]

$$\alpha(\mathbf{X}) = \prod_k [1 - \upsilon(k, \mathbf{X})] \tag{4}$$

where the product is over all the frames. This is the *visible-opacity constraint* and it plays a central role in the depth carving process. The classification of opaque

and free space points follows directly from the constraint (Fig. 1): a point is classified as opaque if it is occluded from all the reference frames; and classified as free space if it is visible in at least one reference frame. Of course in practice we cannot realistically expect to be able to view every free space point. However, the only consequence is that those points will be mis-classified as opaque (they will not be carved away), which for view interpolation is not so critical - we would not expect to be able to interpolate parts of a scene that are occluded from all of the reference frames.

The above constraint therefore provides a means of determining opacity from visibility. Although we do not have direct access to the latter, we can however obtain initial approximations using the local correspondence matching about the reference frames. These provide an indication of likely visible surface points along viewing rays, albeit with a degree of ambiguity as discussed earlier, and hence can be used to determine initial visibility estimates. The above relationships between opacity and visibility then provide a means of improving the estimates via iterative refinement; we can use the initial visibility estimates to obtain the opacity values from eqn (4) and then use these to obtain updated visibility values from the relationship in eqn (3), and so on, until convergence. In practice this is best achieved using a probabilistic approach and in the next section we describe a sequential Bayesian formulation for the refinement process.

The method has similarities with the space carving algorithm described by Agrawal and Davis [1]. However, they use different constraints based on the likelihood of a visible surface being at a given point, rather than the opacity and visibility properties used in our formulation. Consequently they need to use temporal selection of frames to avoid valid estimates being carved away due to occlusion. In contrast, our approach allows all the frames to be combined for each point, which is simpler and gives increased potential for refinement.

## 4 Probabilistic Depth Carving

The opacity and visibility values are limited to 0 or 1 and so we treat their estimation as a binary decision problem and obtain solutions using a Bayesian probabilistic formulation. Specifically, we use iterative refinement to update the probability that a point $\mathbf{X}$ is opaque given a measure of how well probabilities for visibility in all the frames support the visible-opacity constraint. This can be formulated as a sequential Bayesian update [4] in which previous estimates of the probability form the prior and the likelihood is given by a constraint support measure, ie for $n > 0$

$$P_{n+1}(\alpha = 1) \;=\; \frac{R_n \; P_n(\alpha = 1)}{R_n \; P_n(\alpha = 1) + (1 - R_n) \; (1 - P_n(\alpha = 1))} \tag{5}$$

where we have omitted the dependence on the point $\mathbf{X}$ for ease of notation. The likelihood $R_n$ given the current state follows from eqn (4)

$$R_n \;=\; \prod_k \left[1 - P_n(\upsilon_k = 1)\right] \;=\; \prod_k P_n(\upsilon_k = 0) \qquad n > 0 \tag{6}$$

where $\upsilon_k \equiv \upsilon(k, \mathbf{X})$ and the product is over all of the reference frames. The iterative process in eqn (5) therefore promotes depths which support the visible-opacity constraint and penalises those that do not.

The above iterative process requires an expression for the probability that a point is occluded in a given frame, ie $P_n(\upsilon_k = 0)$, and suitable initial conditions. Given probabilities for the opacity at the $n$th iteration we base the former on eqn (3): along a viewing ray, points beyond a point with a high probability of being opaque should have high probabilities of being occluded. Hence we use the following expression for the probability of occlusion

$$P_n(\upsilon_k = 0) \;=\; [1 - P_n(\alpha = 1)]f(\mathbf{X}) \;+\; P_n(\alpha = 1) \qquad n > 0 \tag{7}$$

where $f(\mathbf{X})$ is a monotonic function which tends to 1 if $\mathbf{X}$ is likely to be occluded and to 0 if it is likely to be visible. We use the following form for this function

$$f(\mathbf{X}) = 1 - \exp(-s^2/2\sigma_f^2) \qquad\qquad s = max\{\Lambda_i\}/max\{\Lambda\} \tag{8}$$

where $\Lambda$ and $\Lambda_i$ are the set of opacity probabilities for all points along the ray and all points in front of $\mathbf{X}$, respectively. Thus, $P_n(\upsilon_k = 0) \rightarrow 1$ if $\mathbf{X}$ is preceded by a point with a relatively high probability of being opaque and tend to $P_n(\alpha = 1)$ otherwise, ie for visible points it assumes the same (low) probability of opacity. The variance term $\sigma_f^2$ controls the fall off between these two cases.

The required initial conditions are $P_0(\alpha = 1)$ and $P_0(\upsilon_k = 0)$ along each viewing ray for each reference frame. We set $P_0(\alpha = 1) = 0.5$ for all points, ie points are equally likely to be opaque or in free space, and obtain the initial probabilities for occlusion using eqn (7), replacing the opacity probabilities with similarities derived from local correspondence matching about each reference frame as discussed in the previous section. For a point $\mathbf{X}$, we obtain a matching similarity $m(k, \mathbf{X})$ with respect to frame $k$ based on local correspondences within a set of nearby frames $\Gamma_k$ as follows

$$m(k, \mathbf{X}) = \exp(-med\{[I_k(\mathbf{x}_k) - I_j(\mathbf{x}_j)]^2, j \in \Gamma_k\}/2\sigma_m^2) \tag{9}$$

where $med\{\}$ denotes the median operation and $I_k(\mathbf{x}_k)$ is the intensity value in frame $k$ at point $\mathbf{x}_k$ corresponding to the projection of $\mathbf{X}$ (in practice we interpolate the intensity from surrounding pixels). We use the median of the squared intensity differences here to give a degree of robustness and $\sigma_m$ is set to reflect the expected variance between nearby frames. This completes the description of the depth carving algorithm; a summary is given in Fig. 2.

## 5 Multiresolution Implementation

In practice we need to implement a discrete version of the above process, based on a finite number of discrete 3-D points and using discrete reference frames. An efficient means of doing this is to maintain a separate opacity function for each reference frame along with its visibility function, ie associated with a given

(Carry out each step for each 3-D point. Initialise $n = 0$.)

1. Initialise the opacity probabilities so that $P_0(\alpha = 1) = 0.5$.
2. Initialise the occlusion probabilities $P_0(\upsilon_k = 0)$ using eqns (7), (8) and (9), with $m(k, \mathbf{X})$ replacing $P_n(\alpha = 1)$ in eqn (7).
3. Update opacity probabilities $P_{n+1}(\alpha = 1)$ using eqns (5) and (6).
4. Update occlusion probabilities $P_{n+1}(\upsilon_k = 0)$ using eqns (7) and (8).
5. Terminate if converged; otherwise increment $n$ and goto to step 3.

**Fig. 2.** The probabilistic depth carving algorithm.

frame $k$ are opacity values $\alpha(k, \mathbf{X}_i)$ and visibility values $\upsilon(k, \mathbf{X}_i)$, where the discrete 3-D points $\mathbf{X}_i$ are defined along each of the pixel viewing rays. This enables efficient updating of the occlusion probabilities using the opacities along the viewing rays as required in step 4 of Fig. 2. The combination of visibility values to update the opacity in step 3 is then achieved by interpolating amongst the closest 3-D points within the visibility function of each reference frame. Following depth carving, the result is therefore a set of opacity and visibility functions with respect to each reference frame.

We also use a multiresolution implementation to improve computational efficiency by directing the depth sampling towards significant areas such as in the vicinity of surface boundaries. This is done using a course-to-fine focusing strategy. We first generate levels of a Gaussian pyramid for each frame [7]. Starting at the coarse resolution level, we compute matching similarities for each reference frame and each viewing ray at a small number of discrete points within a pre-determined depth range as in eqn (9). Depth carving then proceeds as detailed in Fig. 2 for each 3-D point. The procedure is then repeated for the next higher resolution level of the pyramid using an increased number of sample points. This is achieved by defining a set of 'child points' along the viewing rays corresponding to the 4 child nodes in the Gaussian pyramid for each 'parent point' in the previous level, with the depths of the child points evenly distributed about that of the parent. Carving then proceeds for the child points but excluding those for which consistent opacity classification was achieved at the parent level. The opacity for these points is set to that of the parent point. We define a parent point to be consistent if its opacity classification matches that of all neighbouring points along adjacent pixel viewing rays. In other words, we propagate down consistent opacity values, hence avoiding depth carving within the corresponding depth ranges at the next level. This course-to-fine focusing then continues until the highest resolution level is reached.

## 6 Depth Carving Examples

Examples illustrating the depth carving process are shown in Figs. 3 and 4. Reference frames from two sequences are shown in Figs. 3a and 4a: the first is a

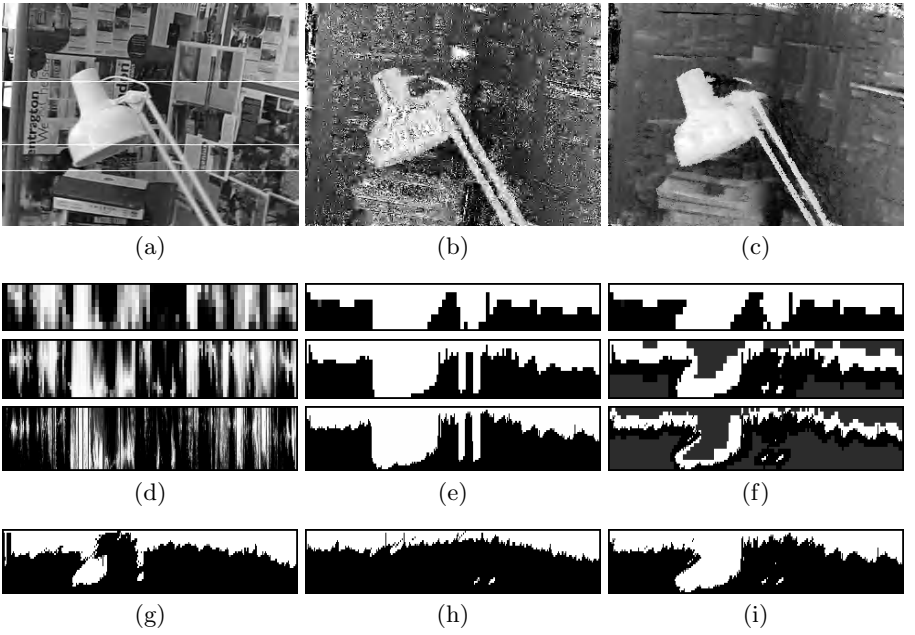


**Fig. 3.** Multiresolution depth carving for a reference frame in the desk sequence.

desk scene with 76 frames in which the camera translates and rotates horizontally whilst keeping the lamp roughly in centre view; and the second is a garden scene with 161 frames in which the camera moves around an ornament by approximately 180 degrees whilst keeping it roughly in centre view. Both sequences contain significant occlusion and the garden sequence contains widely disparate views with very few common features across all the frames. We calibrated the sequences using the recursive structure from motion algorithm described in [3] to obtain relative metric 3-D camera positions for each frame and sparse depth estimates corresponding to tracked feature points. The latter were obtained using the KLT tracker [16]. For the garden sequence we processed the frames in three sections due to features moving out of view.

The results of the multiresolution depth carving process for the desk sequence can be seen in Figs. 3d-i. We used 16 reference frames each with 20 neighbouring frames to obtain the local matching similarities. We set the matching and visibility parameters $\sigma_m$ and $\sigma_f$ as described in [18]. The plots in Fig. 3d show the matching similarities obtained along the middle scanline shown in Fig. 3a at the sample depths on each level of the Gaussian pyramid, where the closest depth is at the bottom of the plot. Figures 3e-f show the visibility and opacity values respectively for the same scanline and depths following convergence of the carving process. The shaded areas in the opacity plots indicate that opacity values are propagated down from the previous level and Fig. 3i shows the final opacity values obtained by combining the levels. Final opacity values for the top and bottom scanlines are shown in Figs. 3g and 3h, respectively. Fig. 3c shows

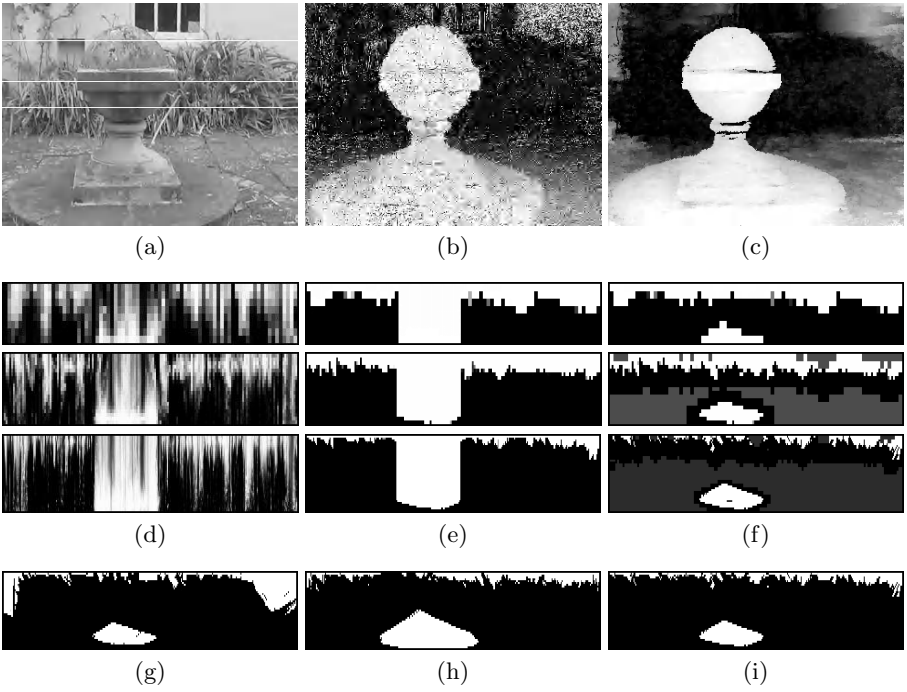



**Fig. 4.** Multiresolution depth carving for a reference frame in the garden sequence; (d-f) and (i) refer to the bottom scanline in (a) and (g,h) refer to the top and middle scanlines, respectively.

the closest depth with opacity probability above a given threshold, ie corresponding to the visible surface points in the frame. For comparison, Fig. 3b shows the depths with the highest local matching similarity. A similar set of results for the garden sequence are given in Fig. 4, where we used 15 reference frames each with 20 neighbouring frames.

The main observation from these results is that despite the high level of ambiguity in the depth estimates obtained from the local correspondence matching (see Figs. 3b,d and 4b,d), the depth carving process correctly determines the opacity classification for the significant parts of the scene. For example, in the desk sequence, large portions of the free space surrounding the lamp shade and stand have been successfully carved away, leaving only those parts not visible in any of the reference frames. Similar comments apply to the results for the garden sequence, with the free space around the ornament being carved away. Note in particular the correct isolation of the lamp stand and cable in the desk sequence and the curved surface carved away for the front of the lamp shade and the ornament. The latter is not apparent in the local depth estimates and this provides a good illustration of how depth carving can combine disparate views to improve estimates of scene structure. Note also from Figs. 3f and 4f how the multiresolution implementation successfully focuses the depth sampling at each level around the key surface areas, thus helping to minimise computational cost.

(a) (b) (c)

**Fig. 5.** View interpolation for intermediate frames in the original sequence: (a) original frame; (b) reconstructed visible surface depths; (c) interpolated view.

Less satisfactory, however, is the amount of noise present in the opacity estimates for the background in the two sequences. This is caused by false foreground depths detected by the initial local correspondences due to textureless regions which are not carved away when combined with other views. However, as noted earlier, in textureless regions such errors are less critical for view interpolation.

## 7 View Interpolation

Having determined opacity and visibility representations for each reference frame, it is straightforward to generate corresponding functions for virtual frames using the visible-opacity constraint in eqn (4). For each sample point along each viewing ray in the virtual frame we interpolate visibility values $v(k, \mathbf{X})$ with respect to each reference frame and then combine them using eqn (4) to give the opacity in the virtual frame. This becomes a computationally inexpensive binary operation if we first make a maximum likelihood assignment of opacity values for the reference frames and is one of the key advantages of the depth carving framework. An example is shown in Fig. 5b which shows the visible surface depth estimates for an intermediate frame in the original sequence between two reference frames. The essential scene structure has clearly been maintained within this reconstructed view. Further examples for virtual frames away from the original camera trajectories are shown in Fig. 6a.

Novel views can be generated from the visible surface depth maps using reprojection. Thus, for pixels in a virtual frame, we collect pixel values from the reference frames, excluding those frames from which the point in question is occluded. These values are then processed to determine the likely value in the new view. This raises the issue of how best to combine pixel values to produce

(a) (b) (c)

**Fig. 6.** Interpolated novel views: (a) reconstructed visible surface depths; (b) interpolated novel views using depth maps in (a); (c) other interpolated views.

realistic views [9,10]. Here, we opted for two simple combination strategies: the median of the collected pixel values; or using the pixel value from the closest reference frame. The former proved marginally better for the desk sequence since it gives a degree of robust averaging which results in better consistency. This works because of the limited variation in lighting effects between the different views. In contrast there are significant light changes in the garden sequence and in this case taking pixel values from the closest (non-occluded) reference frame proved more effective. Figure 5c shows reconstructed versions of the intermediate original frames shown in Fig. 5a based on the visible surface depth maps shown in Fig. 5b. Given the simplicity of the pixel combination algorithm, these results are good and they illustrate clearly the validity of the opacity representation derived from the depth carving. Similar comments apply to the novel views generated away from the original camera paths as shown in Fig. 6, although there are some small ghosting effects and holes appearing in garden sequence, especially around the base of the ornament, due to errors in the depth maps.

## 8 Conclusions

The view interpolation algorithm presented here has several key benefits. It enables the generation of view-centred depth representations without having to establish dense correspondence between reference views. This is useful for sequences containing widely disparate views. The visible opacity model allows the generation of depth representations and novel views without recourse to explicit surface fitting. Significantly, the depth carving allows ambiguities in depth estimates obtained in one view to be resolved using more reliable estimates obtained from different views and relating to different parts of the scene, for example within textured areas. We are not aware of this technique being used before and our

results suggest that it has considerable potential. We are currently investigating the use of local spatial and depth constraints and more sophisticated methods for combining pixels when generating interpolated views.

**Acknowledgement.** The authors are grateful to the Independent Television Commission, UK, for financial assistance.

## References

1. M Agrawal and L.S Davis. A probabilistic framework for surface reconstruction from multiple images. In *Proc Conf on Computer Vision and Pattern Recognition*, 2001.
2. S Avidan and A Shashua. Novel view synthesis in tensor space. In *Proc Conf on Computer Vision and Pattern Recognition*, 1997.
3. A Azarbayejani and A P Pentland. Recursive estimation of motion, structure and focal length. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 17(6):562–575, 1995.
4. J.O Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer, New York, 1985.
5. J.S Bonet and P Viola. Roxels: Responsibility weighted 3d volume reconstruction. In *Proc Int Conf on Computer Vision*, 1999.
6. A Broadhurst, T.W Drummond, and R Cipolla. A probabilistic framework for space carving. In *Proc Int Conf on Computer Vision*, 2001.
7. P.J Burt and E.H Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532–540, 1983.
8. N.L Chang and A Zakhor. Constructing a multivalued representation for view synthesis. *Int Journal of Computer Vision*, 2(45):157–190, 2001.
9. P.E Debevec, C.J Taylor, and J Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proc ACM SIGGRAPH*, 1996.
10. A Fitzgibbon, Y Wexler, and A Zisserman. Image-based rendering using image-based priors. In *Proc Int Conf on Computer Vision*, 2003.
11. S.J Gortler, R Grzeszczuk, R Szeliski, and M.F Cohen. The lumigraph. In *Proc ACM SIGGRAPH*, 1996.
12. R Koch, M Pollefeys, and L Van Gool. Multi viewpoint stereo from uncalibrated video sequences. In *Proc European Conf on Computer Vision*, 1998.
13. K.N Kutulakos and S.M Seitz. A theory of shape by space carving. *Int Journal of Computer Vision*, 3(38):199–218, 2000.
14. P.J Narayanan, P.W Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proc Int Conf on Computer Vision*, 1998.
15. S Seitz and C.R Dyer. Physically-valid view synthesis by image interpolation. In *Proc IEEE Workshop on Representation of Visual Scenes*, 1995.
16. J Shi and C Tomasi. Good features to track. In *Proc Conf on Computer Vision and Pattern Recognition*, 1994.
17. R Szeliski and P Golland. Stereo matching with transparency amd matting. *Int Journal of Computer Vision*, 32(1):45–61, 1999.
18. A Yao and A Calway. Dense 3-d structure from image sequences using probabilistic depth carving. In *Proc British Machine Vision Conference*, pages 211–220, 2003.

# Sparse Finite Elements for Geodesic Contours with Level-Sets*

Martin Weber[1], Andrew Blake[2], and Roberto Cipolla[1]

[1] Department of Engineering, University of Cambridge, UK,
{mw232,cipolla}@eng.cam.ac.uk
http://mi.eng.cam.ac.uk/research/vision/
[2] Microsoft Research, Cambridge, UK
ablake@microsoft.com

**Abstract.** Level-set methods have been shown to be an effective way to solve optimisation problems that involve closed curves. They are well known for their capacity to deal with flexible topology and do not require manual initialisation. Computational complexity has previously been addressed by using *banded* algorithms which restrict computation to the vicinity of the zero set of the level-set function. So far, such schemes have used finite difference representations which suffer from limited accuracy and require re-initialisation procedures to stabilise the evolution. This paper shows how banded computation can be achieved using finite elements. We give details of the novel representation and show how to build the signed distance constraint into the presented numerical scheme. We apply the algorithm to the geodesic contour problem (including the automatic detection of nested contours) and demonstrate its performance on a variety of images. The resulting algorithm has several advantages which are demonstrated in the paper: it is inherently stable and avoids re-initialisation; it is convergent and more accurate because of the capabilities of finite elements; it achieves maximum sparsity because with finite elements the band can be effectively of width 1.

## 1 Introduction

Level-set methods are generally useful for the analysis of image data in 2D and 3D when one has to solve an optimisation problem with respect to an interface [1,2,3,4,5,6,7]. In this paper, we will present a novel numerical scheme to solve the problem of minimising a certain geodesic length in two dimensions which was proposed [8,9] to achieve the attraction to contours in images. For the geodesic model, the cost $C$ (*Riemannian length*) of an interface $\Gamma$ is the integral of a local density $g$ (*Riemannian metric*) over the interface:

$$C = \int_{\Gamma} g \tag{1}$$

* This work was supported by the EPSRC, the Cambridge European Trust and a DAAD-Doktorandenstipendium (Germany).

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 391–404, 2004.
© Springer-Verlag Berlin Heidelberg 2004

where the standard Lebesgue measure is used to integrate the scalar function $g$ over the set $\Gamma$. Differential minimisation of $C$ leads to a *gradient descent* scheme. Level-set methods [1] introduce a level-set function[1] $\phi$ to represent the interface $\Gamma$ implicitly as the zero level-set: $\Gamma := \phi^{-1}(0)$. The implicit representation links $\phi$ (as the introduced *analytic entity*) with the *geometric entity* $\Gamma$: $\phi \mapsto \Gamma(\phi)$ and allows for changes in the topology during the evolution. Furthermore, it was pointed out [10] that this relationship can be made one-to-one by imposing the signed distance constraint. The conceptual advantage is then that $\phi$ is (up to a sign) uniquely determined by $\Gamma$ and that one can also write $\Gamma \mapsto \phi(\Gamma)$. In this way $\phi$ gets the intrinsic geometric meaning as the distance function for $\Gamma$.

### 1.1 Differential Minimisation and Level-Set Evolution

For the evolution, one introduces an evolution parameter $t \in R$ and $\phi$ becomes *time*[2] dependent. One starts with an initial function $\phi(0, .)$ and prescribes an evolution $\phi(t, .)$ that tends towards local minima of the cost $C$ using gradient descent. In the level-set formulation, the gradient descent is expressed in the evolution equation, a *partial differential equation* (PDE):

$$\frac{d\phi}{dt} = \beta \tag{2}$$

where, at the interface $\Gamma$, $\beta$ is the differential of the cost[3]: $\beta|_\Gamma := -\frac{\delta C}{\delta \phi}$ and is defined globally as in [10] to maintain the signed distance constraint. The signed distance constraint is well known for its desirable conceptual and numerical properties [10]. Where $\phi$ is differentiable, we have $|\nabla\phi(x)| = 1$ and, for $x \in \Gamma$, one has particularly simple expressions for the curve's *normal* $N(x) = \nabla\phi(x) \in S^1$ and curvature $\kappa(x) = \nabla^2\phi(x) \in \mathbb{R}$.

### 1.2 Previous Numerical Problems of Evolving Level-Sets

In the following, $u$ denotes the numerical representation of the level-set function $\phi$. There are two major issues in the numerical implementation of the PDE (2): one is efficiency and the other is stability. Potential inefficiency arises from the need to maintain an entire (2D) function $u$ in order simply to obtain the curve $\Gamma$. "Banded" schemes have been suggested [11,2,12] which restrict computations to the immediate neighbourhood of $\Gamma$. Because the interface $\Gamma$ can leave the band, those schemes require an the algorithm to extend the sparse representation $u$ as signed distance map. However, the extension outside the current band is only consistent if the signed distance property is preserved by the evolution [10].

The implementation of the evolution (2) on a grid of pixels in finite difference schemes [1,11,2,3] results in a stability problem, illustrated by the bunching of levels in Figure 1. Although the signed distance constraint used by Gomes and Faugeras [10] maintains the equal spacing of levels in principle, the numerical implementation (discretisation and finite numerical accuracy) still causes a

[1] $\phi$ is a continuous, real valued function

[2] One refers to the parameter $t$ as *time* although it is not related to physical time.

[3] $\frac{\delta}{\delta\phi}$ denotes variational differentiation and $\beta|_\Gamma$ is detailed in (18) for the cost (1).

drift which eventually destroys the equal spacing. The bunching of levels destabilises the evolution and affects the convergence. Therefore, previous methods required a separate re-initialisation procedure in order to restore the signed distance property. One also needs to select a suitable frequency for invoking the re-initialisation procedure to maintain stability of the evolution.

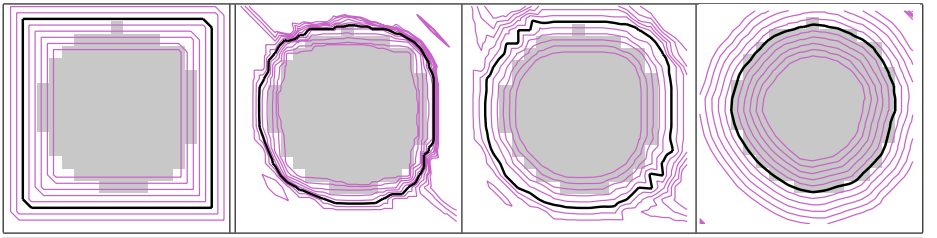



**Fig. 1. Finite element approach has better stability:** The figure compares three different geodesic contour implementations. The initial shape is a square ($18 \times 18$ pixels) and the target shape is a discrete circle (shaded pixels). The zero-level is indicated as a dark line in each case and neighbouring levels are drawn with a level spacing of 0.5 pixel units. (a) initialisation by a rectangle. The following images display the propagation of the level sets when a time step of $\Delta t = 0.1$ is used to evolve the level-set function to $t = 20$. (b) The Hamilton-Jacobi evolution [2] causes a bunching of levels which destabilises the evolution and requires a separate re-initialisation procedure. (c) The signed distance evolution [10] in grid representation improves the stability but still has a slow drift from the signed distance property which also requires a re-initialisation procedure. (d) Novel sparse finite element evolution maintains the signed distance constraint indefinitely, with no need for re-initialisation.

### 1.3 Novel Numerical Solution: The Sparse Finite Element Approach

In this paper, we solve the problems of efficiency and stability by proposing a novel scheme that uses finite elements [13,14] to represent and evolve $u$. Finite elements have been used before in the context of level-set methods: in [15] a finite element scheme is used to represent temperature changes along an interface, while the interface itself is evolved using a finite difference scheme. Preußer and Rumpf [16] work with 3D cubical elements (of mixed polynomial degree) and evolve all levels in the computational domain. Our method introduces a sparse simplicial element representation and combines a weak form of the geodesic evolution equation with the inbuilt preservation of the signed distance constraint:

- The band is represented as a simplicial complex, over which simplices are continually added and deleted, in a fashion which is integrated and harmonious with the differential evolution of $u$. No mode switching is required to deal with the interface $\Gamma$ falling out of the band.
- The simplicial representation of the band allows it to have minimal width, resulting in enhanced efficiency. Derivatives are treated by our weak formu-

lation with no need for conditional operators. As a consequence, no second order derivatives of the level-set function have to be computed explicitly.
– With finite elements, the function $u$ is defined everywhere, not just at grid locations, and sub-grid accuracy is particularly straightforward.
– The signed distance constraint is maintained actively in a stable, convergent fashion, even over indefinite periods of time. This results in an algorithm which is demonstrably more stable (Figure 1) and more accurate (Figure 2) than previous approaches [2,10].

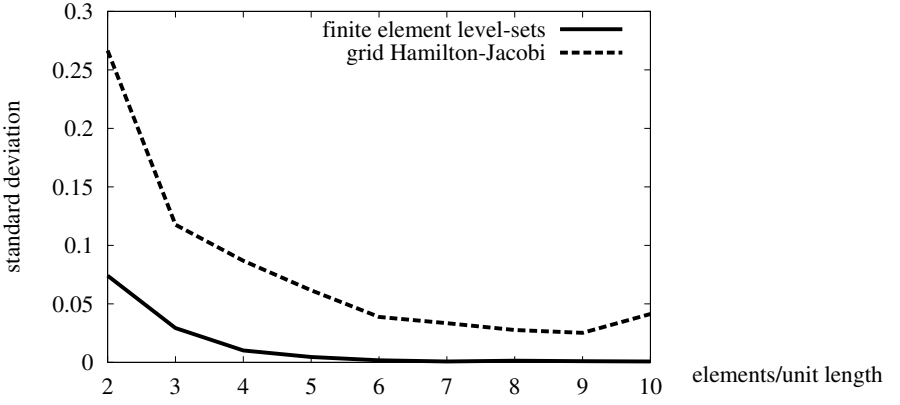



**Fig. 2. Superior accuracy:** The diagram shows deviations of the detected interface from the unit disc. The unit disc is used as target shape of the geodesic evolution (see Figure 1). The diagram shows the deviations (vertical-axis) between the result of the level-set evolution and the target shape when the pixel resolution is varied (horizontal-axis). The lines in the diagram correspond to the Hamilton-Jacobi scheme and the novel method presented in this paper. The new method clearly performs better. The increase in deviation for the grid method on the right is caused by numerical instabilities that occur when no re-initialisation is used.

## 2 Efficient Representation with 'Banded' Finite Elements

The new numerical representation $u$ consists of a *global* and a *local* component:

– The local component inside each element is a polynomial in $d = 2$ variables which prescribes the location of the zero level-set inside the element.
– The global component, the 'band', is a *simplicial complex* that consists of the minimal set of elements that contain the zero level-set (Figure 3). We refer to elements that are members of the complex as being *active*. The representation differs from standard finite element representations in that the complex is sparse and in that it has to be changed dynamically to maintain the containment property for the evolving interface $\Gamma$.

### 2.1 Local Representation: Element Polynomial

Following standard methods found in finite element methods [13,14], we use the standard $d$-simplex to represent $u$ locally as a polynomial of fixed degree $p$ in $d$ dimensions. The standard simplex $T_0^d$ is defined to be the convex hull of the standard Euclidean basis vectors $b_1, b_2, .., b_d \in \mathbb{R}^d$ and the origin $b_0 = 0$. In $d = 2$ dimensions, the standard simplex is simply a triangle as in Figure 3. We adopt the following terminology from finite element methods [13,14]:

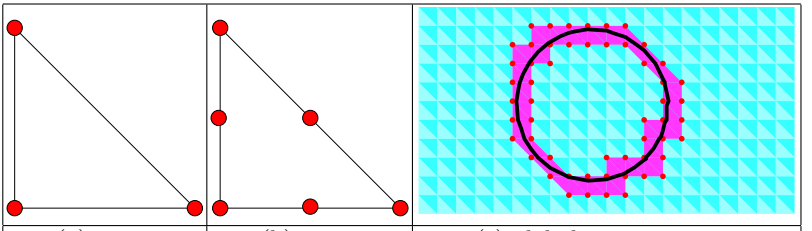



**Fig. 3. Sparse representation (2D):** Inside each simplex, the level-set function $u$ is defined by the values of the nodes. (a) shows the location of nodes within a first order element and in (b) for a second order element. (c) For the global representation, the plane is partitioned into standard simplices (shaded lightly) and computations are restricted to the active complex $\mathcal{A}$ (shaded darker) which consists of the minimal set of elements that contain the zero level-set (in this example a circle).

- a *node* is a location $p_i \in T_0^d$ together with a real value and we position the nodes as indicated in Figure 3 on the grid $\frac{1}{p}\mathbb{Z}^d$.
- the *nodal basis function* $e_i$ associated with node $i$ is the unique [13] polynomial of degree $p$ that evaluates at the nodes to: $\forall j \;\; e_i(p_j) = \delta_{ij}$.
- $u$ is a linear combination of the nodal basis functions: $u = \sum_i \mathbf{u}_i \, e_i$.

The fact that integration is a linear operation will enable us to express all occurring integrals (9),(11) as linear combinations of a few integral constants (15),(20).

### 2.2 Global Representation: Active Simplicial Complex

Our global representation of the functional $u$ consists of the active complex $\mathcal{A}$ covering the area $\Omega$. Each $d-simplex$ of the complex is mapped to a standard element and defines in this way a global functional $u$ on the area $\Omega$. By the sharing of nodes, we obtain a global functional that is automatically continuous[4]. We restrict our current exposition of global representations to the 2-dimensional case. Note however, that our formulation is equally applicable for hyper-surfaces of higher dimensions (e.g. in the 3D case, one can partition the space by choosing the *Delaunay tetrahedrisation* [18] of the node set $\mathbb{Z}^3 \cup \left((\frac{1}{2}, \frac{1}{2}, \frac{1}{2})^\top + \mathbb{Z}^3\right)$ where all tetrahedrons are of the same type). In 2D, a rectangular area (e.g. the image plane) can be partitioned using standard simplices as illustrated in Figure 3(c).

[4] This is a significant advantage over representations that do not enforce continuity (like for instance the *surfel* representation used in [17]).

## 3 Stable Dynamics to Evolve the Novel Representation

Having defined the efficient numerical representation of $u$, we now show how a stable evolution can be defined which is at the heart of our method. In order to avoid re-initialisation procedures, we integrate the signed distance property into the evolution equations by introducing an error functional $r$ which penalises deviations from the desired interface motion $\beta|_\Gamma$ as well as deviations from the signed distance property. The evolution algorithm then minimises this functional.

### 3.1 Components of the Evolution Equations

Firstly, unlike [10], we express the signed distance constraint in the following form:

$$(\nabla_x\, u)^2 - 1 = 0 \tag{3}$$

Secondly, the desire to move the interface at a normal speed $\beta|_\Gamma$ simply implies

$$u_t|_\Gamma = \beta_\Gamma \tag{4}$$

for the update of $u$ by (2). We consider interface motion of the general form [2]

$$\beta|_\Gamma(t, x, N, \kappa) \tag{5}$$

which involves *external forces* by the dependence on $x \in \mathbb{R}^d$ and $t \in \mathbb{R}$ as well as the *intrinsic quantities* orientation $N$ and curvature $\kappa$. Note that this means that $\beta|_\Gamma$ depends on the $2^{\text{nd}}$ derivative of $u$ and that we have $\beta|_\Gamma(t, x, N, \kappa) = \beta|_\Gamma(t, x, \nabla u, \nabla^2 u)$ due to the signed distance constraint. In this paper we apply the general method to the geodesic problem (1) for which $\beta|_\Gamma$ is given by (18).

### 3.2 Discrete Dynamics

Now the evolution of the level-set function is set-up in discrete space and time, in terms of the displacement $v$ of the function $u$ over a time-step $\Delta t$:

$$u(t + \Delta t, .) = u(t, .) + v(t, .). \tag{6}$$

Here $v$ is represented over the finite element basis, in the same way as $u$ is, and represents displacement for a time $\Delta t$ at velocity $\beta$:

$$v = \Delta t\ \beta(u + v) \tag{7}$$

where we have chosen to evaluate $\beta$ at $u+v$ (instead of $u$ in the explicit scheme) to obtain an implicit scheme [19] which does not limit the magnitude of $\Delta t$.

### 3.3 *Weak Formulation* of Evolution Dynamics

Inspired by the *Petrov-Galekin formulation* [13,14] used in finite element methods, we employ a *weak formulation* of (3) and (4) with the following advantages:

- It allows us to measure and use the constraint equations for the entire active area $\Omega$, and not just at discrete, sampled locations [10].
- It allows for curvature dependent interface motion (5) even in the case of first order elements ($p = 1$) by the use of Green's theorem.
- It gives an appropriate regularisation of derivative operators without the need of switch-operators found in grid representations [2,10].

In the Petrov-Galekin form, one uses the nodal basis functions $e_i$ $i \in \{1, ..., n\}$ as *test functions* to measure deviations from the desired evolution properties (3) and (4). First, the signed distance equation (3) becomes a set of equations:

$$z_1^i = 0, \quad \text{for} \quad i = 1, \ldots, n \tag{8}$$

where

$$z_1^i := \int_\Omega \left( (\nabla u + \nabla v)^2 - 1 \right) e_i. \tag{9}$$

Secondly, the velocity law (7) is expressed as

$$z_2^i = 0, \quad \text{for} \quad i = 1, \ldots, n \tag{10}$$

where

$$z_2^i := \int_\Omega (v - \Delta t\, \beta)\, e_i. \tag{11}$$

We now introduce[5] an optimisation problem to determine the update of the level-set function which minimises deviations from (8) and (10).

### 3.4 Level-Set Update Equations as Optimisation Problem

The two sets of equations (9) and (11) represent an overdetermined system of $2n$ equations in $n$ unknowns. We measure the deviations in the following functional:

$$r^2 := |z_1|^2 + \alpha^2 |z_2|^2, \tag{12}$$

where $z_1 = (z_1^1, \ldots, z_1^n)^\top$ and similarly for $z_2$, and $\alpha \in \mathbb{R}_+$ is an arbitrary positive constant that balances the competing terms in the optimisation problem.

The functional can be written compactly by expressing $z_1^i$ and $z_2^i$ in terms of the node values $\mathbf{v} = (v_1, \ldots, v_n)^\top$ for the displacement $v$, and similarly for $u$:

$$z_1^i = \mathbf{u}^\top Q^i \mathbf{u} - k^i + 2\mathbf{u}^\top Q^i \mathbf{v} + h_i(\mathbf{v}) \tag{13}$$

$$z_2^i = P^i\, \mathbf{v} - \Delta t \int_\Omega e_i\, \beta \tag{14}$$

where $h_i(\mathbf{v}) := \mathbf{v}^\top Q^i \mathbf{v}$ and where constants $k, P, Q$ are defined as:

[5] The optimisation problem introduced here is not to be confused with the optimisation problem (1) that gives rise to the differential evolution $\beta|_\Gamma$ in the first place.

$$k^i := \int_\Omega e_i, \quad P_{ab} := \int_\Omega e_a\, e_b, \quad Q^i_{ab} := \int_\Omega \langle \nabla e_a, \nabla e_b \rangle\, e_i \tag{15}$$

The quantities $(k, P, Q)$ can be pre-computed analytically, and stored as constants. Note that the deviation $z_1$ is almost affine in the unknown speed $v$ since the $h_i(\mathbf{v})$ are small, provided $u$ approximates the signed distance property and if the time step $\Delta t$ is sufficiently small. In that case (13) can be linearised, ignoring $h$, by replacing $z_1^i$ by $\tilde{z}_1^i = \mathbf{u}^\top Q^i \mathbf{u} - k^i + 2\mathbf{u}^\top Q^i \mathbf{v}$. We solve the linear least-square problem numerically by using the *conjugate gradient method* [20]. We exploit the sparsity over the computational grid which allows the linear simultaneous equations to be expressed in banded form over the nodal basis. Using the banded form, we solve for $\mathbf{v}$ in $\mathcal{O}(n)$ where $n$ denotes the number of nodes and is proportional to the length of the interface in element units.

### 3.5 How the Global Evolution Ensures the Containment Property

For our method to be viable, we require to detect changes in the active complex $\mathcal{A}$ efficiently. The new method is outlined in Algorithm 1. After each local evolution (lines 2-4 of **evolve**), we adjust the active complex by adding and removing elements (lines 5-14 of **evolve**) to ensure that it contains the current interface $\Gamma$ and that it is minimal with that property. The initialisation of neighbouring elements is well defined; this is the case because although we restrict the *numerical representation* to the sparse complex, $u$ is indeed defined globally by the signed distance constraint. This justifies the use of the extrapolation procedure in **activate()**. Extrapolation[6] is natural since $u$ is represented in full functional form and does not require any separate interpolation mechanism for evaluation.

The maintenance of $\mathcal{A}$ (activation and removal of elements) is performed efficiently in **evolve** by looking at the edges of the complex. Note that the level-set function along an edge is a polynomial of degree $p$ and that it is determined by the $p+1$ nodes located along the edge [13]. Hence, the problem of deciding where $\mathcal{A}$ needs modification reduces to the problem of finding the roots of the edge-polynomial which is straightforward for linear and quadratic elements.

## 4 Geodesic Contour Detection with Sparse Finite Elements

It has been known for some time [8,9] that the problem of contour detection can be cast into the problem of minimising a Riemannian length functional that is induced by the image. In order to define the cost functional $C$ (1), one starts by introducing the *local* measure for edges $g$, which is normalised so that $g(x) \in [0, 1]$. The basic construction adopted here uses a positive *edge detector* function $f$. In this paper, we employ the following monotonic function [19]

$$g := 1 - \exp\left(-\tfrac{a}{|\nabla f_\sigma|^q}\right) \tag{16}$$

[6] For 1st order elements the value $u_c$ in element $(abc)$ is $u_c = u_a + u_b - \bar{u}_c$ where $\bar{u}_c$ is the known value of the node that is obtained by reflecting node $c$ along the edge $ab$.

**Algorithm 1** Geodesic level-set algorithm with sparse finite elements

```
detect geodesic contour:
smoothen image with Gaussian(σ)
compute image metric g {see (16)}
initialise level-set u as sparse finite element complex A
repeat
    evolve(u,g)
until converged
output interface Γ(u)
return
```

```
activate(T):
for all nodes V of element T do
    if V ∉ A then
        initialise V (extrapolate from active T-adjacent elements)
    end if
end for
add element to A
return
```

```
evolve(u,g):
{Update u:}
compute A_1, A_2 and b_1,b_2 such that z_l = A_l v + b_l {see (13), (19)}
solve the least square equation A^⊤(Av + b) = 0 (C.G. method)
u ← u + v {see (6)}
{Now update the active complex A:}
for all edges E ∈ A that contain a root do
    for all E-adjacent elements T ∉ A do
        activate(T)
    end for
end for
for all T ∈ A do
    if no edge of T is active then
        remove T from A
    end if
end for
return
```

where $f_\sigma$ is smoothed with a Gaussian of scale parameter $\sigma$ and $a, q$ are real constants that control the scale of sensitivity and the slope as parameters of the normalising function. For the experiments in this paper, we employ the following edge detector functions $f$:

- **Colour edges:** $f(x) = I(x)$ where the image $I$ has values in rgb-space.
- **Gaussian colour edges:**

$$f(x) = \exp\left(-\tfrac{\gamma}{2}(I(x) - \bar{y})^\top \Sigma^{-1}(I(x) - \bar{y})\right) \tag{17}$$

  here, $\bar{y}$ is a fixed colour, $\Sigma$ is a covariance matrix, $\gamma$ a positive constant and $I$ has values in rgb-space. For our examples, we obtain the covariance matrix by sampling $m$ pixel-values $\{y_j\}$ in a user-defined region:

$$\bar{y} := \tfrac{1}{m}\sum_j y_j \qquad \Sigma := \left(\tfrac{1}{m}\textstyle\sum_j y_j\, y_j^\top\right) - \bar{y}\,\bar{y}^\top$$

### 4.1 Geodesic Evolution Equation

The velocity function $\beta$ which asymptotically minimises $C$ for signed distance functions can be shown [12,21] to be:

$$\beta|_\Gamma = \langle \nabla g, \nabla \phi \rangle + g\left(\nabla^2 \phi + c\right) \tag{18}$$

where $c$ is the coefficient of the so-called "balloon force" [22,3,12] and we refer to [7] for the connection to parametric snakes [23] that has been discussed in the literature. The "balloon force" term does not arise from the cost minimisation but is useful in certain situations [21].

### 4.2 Numerical Form of the Evolution

In order to apply the new numerical method of Section 3, we have to perform the evaluation of $z_2$ in (14) for the geodesic case. It is well known ([19], *implicit scheme*) that the evaluation of $\beta(u+v)$ instead of $\beta(u)$ in (14) is preferable since it does not impose a limit on the time-step $\Delta t$. Note that unlike in the case of the diffusion on images [19], the implicit scheme is computationally feasible due to the sparse representation. Using Green's formula, it can be shown that the weak form (14) of the velocity equation (18) becomes:

$$z_2^i = \Big(P_i - \Delta t\,[(B_i - Q_i)\mathbf{g}]^\top\Big)\mathbf{v} - \Delta t\,\big(c\,P_i\,\mathbf{g} + \mathbf{u}^\top (B_i - Q_i)\,\mathbf{g}\big) \tag{19}$$

where we have also represented $g$ in nodal basis and where we have introduced

$$B_{ik}^j := \int_{\partial\Omega} e_k\; e_i\; \langle \nabla e_j, \mathcal{V}\rangle \tag{20}$$

as boundary integral constants with $\mathcal{V}$ denoting the outward normal along $\partial\Omega$.

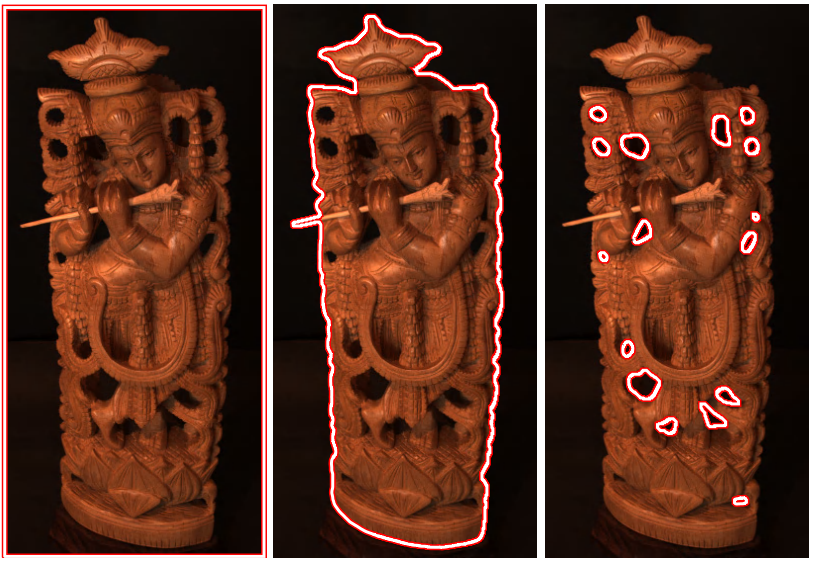
*Proof.* (Outline) by (18) the interface speed is $\beta|_\Gamma = \operatorname{div}(g\,N) + c\,g$ and, by the signed distance property $N = \nabla u$ in the explicit scheme and $N = \nabla(u+v)$ in the implicit scheme. Inserting $\beta$ into (14) one obtains the following non-trivial term in the expression for $z_2^i$: $-\Delta t \int_\Omega \operatorname{div}(g\,N)\,e_i$. Using Green's formula, we can move one differentiation onto the test function $e_i$ and hence trade the second order derivatives for a boundary integral:

$$\int_\Omega \operatorname{div}(g\,N)\,e_i = \int_{\partial\Omega} e_i\,g\,\langle N, \mathcal{V}\rangle - \int_\Omega g\,\langle N, \nabla e_i\rangle$$

now (19) follows by writing $u = \sum_j \mathbf{u}_j\,e_j$ and $g = \sum_k \mathbf{g}_k\,e_k$. □

## 5 Results

This Section presents experimental results (Figures 4, 5, 6) obtained with our method using first order elements ($p = 1$). The smoothing constant $\sigma$ and the type of metric are selected manually for each example. In order to complete the definition of the metric we determine the parameters $a$ and $q$ in (16) automatically such that the average gradient magnitude $\langle|\nabla f_\sigma|\rangle$ over the image results in $g = \frac{1}{2}$ and that the slope of $g$ with respect to the gradient magnitude equals $-1/\langle|\nabla f_\sigma|\rangle$ at this point. Numerically, we compute the gradient using central difference operators and define the finite element functional $\mathbf{g}$ by introducing one node per pixel. In this way $g$ is defined over the entire image domain (unlike



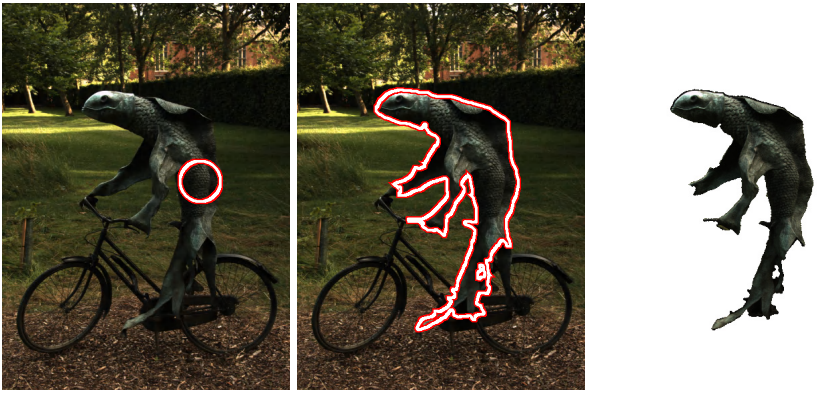
(a) (b) (c)

**Fig. 4. Nested contour detection:** (a) The outline of an image of a wooden Krishna-figure ($266 \times 560$ pixels) is used to initialise the level-set function and to extract a Gaussian distribution of the background. (b) Geodesic minimisation (parameters: $\gamma = 0.1, \sigma = 2, c = 0.4$) leads to the detection of the outer contour. Note the faithful capture of sculptural details such as the pipe which require a stabilised method. (c) Using our metric inversion method [21] the nested contours (holes) are detected automatically.

the sparse functional $u$). For the evolution, we choose the balloon force constant $c$ individually and set $\alpha = 1$ in (12). In our experiments we also set $\Delta t = 1$. In principle, one could choose a larger time step in the implicit scheme but we limit $\Delta t$ here to ensure a small value of $h$ in (13) and not to overshoot any details during the evolution.

## 6 Conclusions and Future Work

We have proposed a new level-set scheme and algorithm for automatically fitting image contours, robustly and efficiently. Even nested contour structures are detected automatically by applying the *metric-inversion* method [21] to the algorithm. We exploited the signed distance constraint systematically to obtain a sparse representation while having a well defined global continuation. For the numerical representation, we replaced previously used finite difference methods and use a dynamically changing finite element complex instead. We incorporated the signed distance constraint into the evolution equations and obtained an algorithm that avoids the periodic re-initialisations required by others. We demonstrated the resulting improvements with respect to stability and accuracy.



(a) (b) (c)

**Fig. 5. Cycling fish sculpture:** (a) A user-specified circle is placed on the image ($429 \times 577$ pixels) as initial level-set and to define a Gaussian colour distribution. (b) Geodesic evolution (parameters: $\gamma = 0.1, \sigma = 1.5, c = -0.3$) with an 'inflating' balloon-force results in the displayed contour. (c) shows the pixels for which $u < 0$.

(a) (b) (c)

**Fig. 6. Cliff example:** The input image ($384 \times 512$ pixels) is displayed in (a) with the user defined initial level-set superimposed. (b) shows the converged contour and (c) the obtained segmentation. In order to define the metric $g$ , a Gaussian in rgb-space that represents the colour distribution inside the circle was defined ($\sigma = 1.5, \gamma = 10^{-1.5}$). A negative balloon-force (c= -0.3) was employed to 'inflate' the initial circle towards the boundaries of the region.

The efficiency, in common with previous schemes, derives from the banded representation, and this is enhanced by the introduction of finite elements which minimises the band width. Using a weak formulation of the evolution equations, we were able to accurately implement curvature dependant evolutions without having to explicitly compute second order derivatives. Various further developments and investigations are underway:

- Extension to the 3D case with tetrahedral elements. As mentioned in Section 2.2, 3-space can be partitioned using a regular mesh of tetrahedrons with a single element shape. Applications include medical imaging and surface reconstruction for model acquisition.
- Implementation of efficient second order finite elements.
- Applications using more sophisticated metrics (e.g. texture segmentation).

## References


1. Osher, S., Sethian, J.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. J. of Comp. Phys. **79** (1988) 12–49
2. Sethian, J.: Level Set Methods. Cambridge University Press, Cambridge (1999)
3. Sapiro, G.: Geometric Partial Differential Equations and Image Processing. Cambridge University Press (2001)
4. Yezzi, A., Soatto, S.: Stereoscopic segmentation. In: Proc. IEEE Int. Conf. on Computer Vision. Volume I. (2001) 56–66
5. Faugeras, O.D., Keriven, R.: Complete dense stereovision using level set methods. In: Proc. European Conf. on Computer Vision. LNCS 1406, Springer (1998) 379–393
6. Osher, S., Paragios, N.: Geometric Level Set Methods in Imaging Vision and Graphics. Springer, New York (2003)
7. Kimmel, R.: Curve Evolution on Surfaces. PhD thesis, Dept. of Electrical Engineering, Technion, Israel (1995)
8. Caselles, V., Kimmel, R., Sapiro, G.: Geodesic active contours. In: Proc. IEEE Int. Conf. on Computer Vision. (1995) 694–699
9. Kichenassamy, S., Kumar, A., Olver, P., Tannenbaum, A., Yezzi, Jr., A.: Gradient flows and geometric active contour models. In: Proc. IEEE Int. Conf. on Computer Vision. (1995) 810–815
10. Gomes, J., Faugeras, O.: Reconciling distance functions and level sets. Journal of Visual Communication and Image Representation **11** (2000) 209–223
11. Malladi, R., Sethian, J., Vemuri, B.: Shape modeling with front propagation: A level set approach. IEEE Trans. Pattern Analysis and Machine Intelligence **17** (1995) 158–175
12. Goldenberg, R., Kimmel, R., Rivlin, E., Rudzsky, M.: Fast geodesic active contours. IEEE Trans. Image Processing **10** (2001) 1467–1475
13. Zienkiewicz, O., Morgan, K.: Finite Elements & Approximation. John Wiley & Sons, NY (1983)
14. Johnson, C.: Numerical Solution of Partial Differential Equations by the Finite Element Method. Cambridge University Press (1987)
15. Ji, H., Chopp, D., Dolbow, J.E.: A hybrid extended finite element/level set method for modeling phase transformations. International Journal for Numerical Methods in Engineering **54** (2002) 1209–1233
16. Preußer, T., Rumpf, M.: A level set method for anisotropic geometric diffusion in 3D image processing. SIAM J. on Applied Math. **62(5)** (2002) 1772–1793
17. Carceroni, R., Kutulakos, K.: Multi-view scene capture by surfel sampling: From video streams to non-rigid 3D motion, shape and reflectance. In: Proc. IEEE Int. Conf. on Computer Vision. (2001) II: 60–67

18. Edelsbrunner, H.: Geometry and Topology for Mesh Generation. Cambridge University Press, Cambridge (2001)
19. Weickert, J., ter Haar Romeny, B., Viergever, M.: Efficient and reliable schemes for nonlinear diffusion filtering. IEEE Trans. Image Processing **7** (1998) 398–410
20. Schwarz, H.: Numerische Mathematik. Teubner, Stuttgart (1993)
21. Weber, M., Blake, A., Cipolla, R.: Initialisation and termination of active contour level-set evolutions. In: Proc. IEEE workshop on Variational, Geometric and Level Set Methods in Computer Vision. (2003) 161–168
22. Cohen, L.: On active contour models and balloons. Computer Vision, Graphics and Image Processing **53** (1991) 211–218
23. Cipolla, R., Blake, A.: The dynamic analysis of apparent contours. In: Proc. IEEE Int. Conf. on Computer Vision. (1990) 616–623

# Hierarchical Implicit Surface Joint Limits to Constrain Video-Based Motion Capture

Lorna Herda, Raquel Urtasun, and Pascal Fua*

Computer Vision Laboratory
EPFL
CH-1015 Lausanne, Switzerland
pascal.fua@epfl.ch

**Abstract.** To increase the reliability of existing human motion tracking algorithms, we propose a method for imposing limits on the underlying hierarchical joint structures in a way that is true to life. Unlike most existing approaches, we explicitly represent dependencies between the various degrees of freedom and derive these limits from actual experimental data.

To this end, we use quaternions to represent individual 3 DOF joint rotations and Euler angles for 2 DOF rotations, which we have experimentally sampled using an optical motion capture system. Each set of valid positions is bounded by an implicit surface and we handle hierarchical dependencies by representing the space of valid configurations for a child joint as a function of the position of its parent joint.

This representation provides us with a metric in the space of rotations that readily lets us determine whether a posture is valid or not. As a result, it becomes easy to incorporate these sophisticated constraints into a motion tracking algorithm, using standard constrained optimization techniques. We demonstrate this by showing that doing so dramatically improves performance of an existing system when attempting to track complex and ambiguous upper body motions from low quality stereo data.

## 1 Introduction

Even though many approaches to tracking and modeling people from video sequences have been and continue to be proposed [1,2,3], the problem remains far from solved. This in part because image data is typically noisy and in part because it is inherently ambiguous [4]. Introducing valid joint limits is therefore one important practical step towards restricting motion tracking algorithms to humanly feasible configurations, thereby reducing the search space they must explore and increasing their reliability.

This is currently done in many existing vision systems [5,6,4,7] but the limits are usually represented in an oversimplified manner that does not closely correspond to reality. The most popular approach is to express them in terms of

* This work was supported in part by the Swiss National Science Foundation.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 405–418, 2004.
© Springer-Verlag Berlin Heidelberg 2004

hard limits on the individual Euler angles used to parameterize joint rotations. This accounts neither for the dependencies between angular and axial rotations in ball-and-socket joints such as the shoulder joint nor those between separate joints such as the hip and knee. In other words, how much one can twist one's arm depends on its position with respect to the shoulder. Similarly, one cannot bend one's knee by the same amount for any configuration of the hip. An additional difficulty stems from the fact that experimental data on these joint limits is surprisingly sparse: medical text books typically give acceptable ranges in a couple of planes but never for the whole configuration space [8], which is what is really needed by an optimization algorithm searching that space.

In earlier work, we proposed a quaternion-based model approach to representing the dependencies between the three degrees of freedom of a ball-and-socket joint such as the shoulder [9]. It relies on measuring the joint motion range using optical motion capture, converting the recorded values to joint rotations encoded by a coherent quaternion field, and, finally, representing the subspace of valid orientations as an implicit surface. Here, we extend it so that it can also handle coupled joints, which we treat as parent and child joints. We represent the space of valid configurations for the child joint as a function of the position of the parent joint.

A major advantage of this quaternion representation it that it provides us with a rigorous distance measure between rotations, and thus supplies the most natural space in which to enforce joint-angle constraints by orthogonal projection onto the subspace of valid orientations [10]. Furthermore, it is not subject to singularities such as the "Gimbal lock" of Euler angles or mapping rotations of $2\pi$ to zero rotations. As a result, it becomes easy to incorporate these sophisticated constraints into a motion tracking algorithm using standard constrained optimization techniques [11].

We chose the case of shoulder and elbow joints to validate our approach because the shoulder is widely regarded as the most complex joint in the body and because position of the arm constrains the elbow's range of motion. We developed a motion capture protocol that relies on optical motion capture data to measure the range of possible motions of various subjects and build our implicit surface representation. We then used it to dramatically improve the performance of an existing system [12] when attempting to track complex and ambiguous upper body motions from low quality stereo data.

In short, the method we propose here advances the state-of-the-art because it provides a way to enforce joint limits on swing and twist of coupled joints while at the same time accounting for their dependencies. Such dependencies have already been described in the biomechanical litterature [13,14] but using the corresponding models requires estimating a large number of parameters, which is impractical for most Computer Vision applications. Our contribution can therefore be understood as a way of boiling down these many hard-to-estimate parameters into our implicit surface representation, that can be both easily instantiated and used for video-based motion capture. Furthermore, the framework

we advocate is generic and could be incorporated into any motion-tracking approach that relies on minimizing an objective function.

In the remainder of the paper, we first briefly review the state of the art. We then introduce our approach to experimentally sampling the space of valid postures that the shoulder and elbow joints allow and to representing this space in terms of an implicit surface in Quaternion space. Finally, we demonstrate our method's effectiveness for tracking purposes.

## 2 Related Approaches

The need to measure joint limits arises most often in the field of physiotherapy and results in studies such as [15] for the hip or [16,8] for the shoulder. Many of these empirical results have subsequently been used in our community.

### 2.1 Biomedical Considerations

When we refer to the shoulder joint, we actually mean the gleno-humeral joint, which is the last joint in the shoulder complex hierarchy. It is widely accepted that modeling it as a ball-and-socket joint, which allows motion in three orthogonal planes, approximates its motion characteristics well enough for visual tracking purposes [3]. This approximation has been validated by a substantial body of biomechanical research that has shown that, because of large-bone-to-skin displacements, no clavicular of scapular motions can be recovered using external markers [17,18].

However, the dependency between arm twist and arm orientation, or swing, is a direct consequence of the complex joint geometry of the shoulder complex [19]. Coupling between elbow and shoulder is not only due to anatomical reasons, but also to the physical presence of the rest of the body, namely the thorax and the head, that limit the amount of elbow flexion for certain shoulder rotations. As to elbow twist, the dependency is anatomical and the available range of motion is directly linked to shoulder orientation [20]. It is those intra- and inter-joint dependencies that make the shoulder and elbow complex ideal to validate our approach. Furthermore, similar constraints exist for the hip and knee joints and our proposed approach should be easy to transpose.

Of course, the interdependence of these joint limits has been known for a long time and sophisticated models have been proposed to account for them, such as those reported in [13,14]. However, the former involves estimating over fifty elastic and viscous parameters, which may be required for precise biomedical modeling, but is impractical for Computer Vision applications and the latter focuses in motions in the sagittal plane as opposed to fully 3–D dimensional movements.

It is worth noting that inter-subject variance has been shown to be extremely small at the shoulder joint level [20]. The online documentation for the Humanoid Animation Working Group confirms that the difference in range of motion of women over men is minimal at the shoulder joint level, and small for the elbow

joint. The experimental data we present in Section 3 confirms this. Thus, it is acceptable to generalize results obtained on the basis of measurements carried out on a very small number of subjects, as we have done in our case, where data collection was carried out on three subjects, two females and one male.

### 2.2 Angular Constraints and Body Tracking

The simplest approach to modeling articulated skeletons is to introduce joint hierarchies formed by independent 1-Degree-Of-Freedom (DOF) joints, often described in terms of Euler angles with joint limits formulated as minimal and maximal values. This formalism has been widely used [5,6,2,4,7], even though it does not account for the coupling of the intra- or inter-joint limits and, as a result, does not properly account for the 3-D accessibility space of real joints.

Furthermore, Euler angles suffer from an additional weakness known as "Gimbal lock". This refers to the loss of one rotational degree of freedom that occurs when a series of rotations at 90 degrees is performed, resulting in the alignment of the axes [21,22]. The swing-twist representation, exponential map, and three-sphere embedding are all adequate to represent rotations and do not exhibit such flaws[23]. However, only quaternions are free of singularities [10]. Furthermore, because there is a natural distance between rotations in quaternion space, it is also the most obvious space for enforcing joint-angle constraints by orthogonal projection onto the subspace of valid orientations. These properties have, of course, been recognized and exploited in our field for many years [24, 25].

The joint limits representation we propose can therefore be understood as a way of encoding the workspace of the human upper arm positions using a formalism that could be applied to any individual joint, or set of coupled joints, in the human body model.

## 3 Measuring and Representing Shoulder and Elbow Motion

We will consider the set of possible joint orientations and positions in space as a path of referential frames in 3-D space [26]. In practice, we represent rotations by the sub-space of unit quaternions $S^3$ forming a unit sphere in 4-dimensional space. Any rotation can be associated to a unit quaternion but we need to keep in mind that the unitary condition needs to be ensured at all times. A rotation of $\theta$ radians around the unit axis $v$ is described by the quaternion:

$$q = [q_x, q_y, q_z, q_w]^T = [sin(\frac{1}{2}\theta)v, cos(\frac{1}{2}\theta)]^T$$

Since we are dealing with unit quaternions, the fourth quaternion component $q_w$ is a dependent variable and can be deduced, up to a sign, from the first three. Given data collected using optical markers, we obtain a cloud of 3-D points by keeping the spatial or $(q_x, q_y, q_z)$ coordinates of the quaternion. In other

words, these three numbers serve as the coordinates of quaternions expressed as projections on three conventional Cartesian axes.

Because we simultaneously measure swing and twist components, and because the quaternion formalism lets us express both within one rotation, this representation captures the dependencies between swing and twist. In this manner, we will have generated joint limits on the basis of motion capture. We will then be able to make use of these joint limits as constraints for tracking and pose estimation, by eliminating all invalid configurations.

### 3.1 Motion Measurement

We captured shoulder and elbow motion using the Vicon$^{TM}$ System, with a set of strategically-placed markers on the upper arm as shown in Figure 1(a). An additional marker is placed at neck level to serve as a fixed reference.

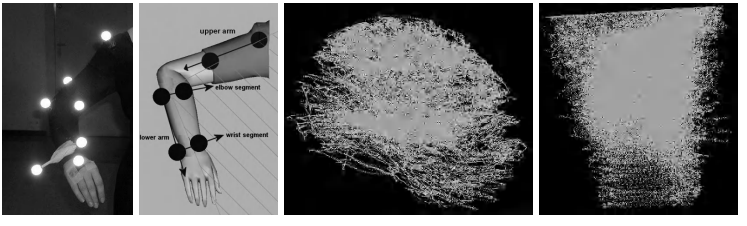
If we wish our joint limits to be as precise as possible, and to reflect the range of motion as closely as possible, we need to pay attention to sampling the space of attainable postures not only homogeneously, but also densely. In general, such a capture sequence lasts several minutes, so that we may obtain such a data set.

### 3.2 Motion Representation

To compute our quaternion field, each joint orientation is first converted to a $3 \times 3$ matrix $M$, where, using Euler's theorem, $M$ may be expressed in terms of its lone real eigenvector $\hat{\mathbf{n}}$ and the angle of rotation $\theta$ about that axis. This in turn may be expressed as a point in quaternion space, or, equivalently, a point on a three-sphere $S^3$ embedded in a Euclidean 4D space. The identification of the corresponding quaternion follows immediately from

$$q(\theta, \hat{\mathbf{n}}) = (\cos\frac{\theta}{2}, \hat{\mathbf{n}}\sin\frac{\theta}{2}) \tag{1}$$

up to the sign ambiguity between the two equivalent quaternions $q$ or $-q$, which correspond to the exact same rotation matrix $M$. When referring to quaternion data, we will from here on always assume that its scalar component is positive.




(a) (b) (c) (d)

**Fig. 1.** Marker positions and associated referentials. (a) Motion capture actor with markers. (b) Shoulder and elbow coordinate frame. (c) Quaternion shoulder data. (d) Elbow coordinate frame.

For each recorded position, we construct a rotating co-ordinate frame for the shoulder joint. As shown in Fig. 1(b), the first axis of the frame corresponds to the line defined by the shoulder and upper arm markers. The second axis is the normal to the triangle whose vertices are the upper arm, elbow and forearm markers. The corresponding plane represents axial rotation and the third axis is taken to orthogonal to the other two. The orientation of each frame is then converted into a quaternion, yielding the volumetric data depicted by Fig. 1(c). For the elbow, we also use the configuration depicted by Fig. 1(b) and transform all marker positions from the global referential to the local shoulder joint referential. Since the elbow has only two degress of freedom, in Fig. 1(d), we represent the resulting data in terms of its two Euler angles.

## 4 Hierarchical Implicit Surface Representation of the Data

Given the volumetric data of Fig.1(c,e), we approximate it as an implicit surface. This will provide us with a smooth and differentiable representation of the space of allowable rotations and its associated metric, which we will use in Section 5 to enforce the corresponding constraints in a very simple manner. This is important because, having been produced by people instead of robots, this data is very noisy. In particular, the regions of lower point density often correspond to motion boundaries and therefore to uncomfortable positions.

### 4.1 Fitting an Implicit Surface

As in [27], given a set of spherical primitives of center $S_i$ and thickness $e_i$, we define an implicit surface as

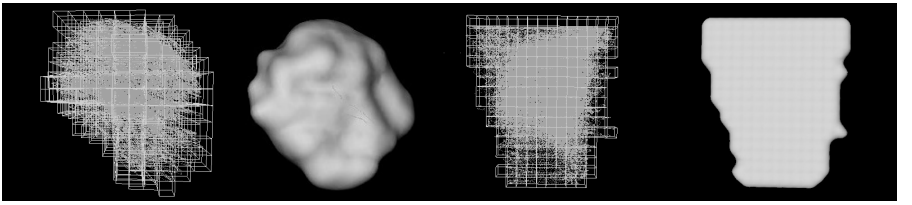
$$S = \{P \in \Re^3 | F(P) = iso\} \tag{2}$$

where

$$F(P) = \sum_{i=1}^{n} f_i(P) \text{ and } f_i(P) = \begin{cases} -kd(P,S_i) + k_i e_i + 1 \text{ if } d(P,S_i) \in [0, e_i] \\ \frac{1}{4}\left[k(d(P,S_i) - e_i) - 2\right]^2 \text{ if } d(P,S_i) \in [e_i, R_i] \\ 0 \text{ elsewhere} \end{cases},$$

where $d$ is the Euclidean distance, $R_i = e_i + \frac{2}{k_i}$ is the radius of influence, $iso$ controls the distance of the surface to the primitives' centers and $k$ its blending properties.

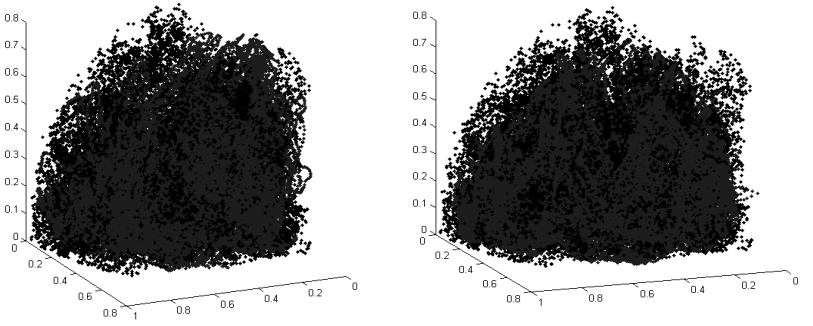
The properties of implicit surfaces and their field functions being the same in 2 and 3–D, we apply the same fitting procedure to the 2–D data for the elbow joint as the 3–D data for the shoulder joint. We begin by voxelizing our space and assigning a point density to each voxel, in very much the same way the Marching Cubes algorithm does for the vertices of its voxels[28]. We then sub-divide the voxels until each voxel has a point density higher than a given threshold, which can be, for example, the density of the data around the center of mass.



(a) (b) (c) (d)

**Fig. 2.** Joint limits for the shoulder and elbow joints. (a) Voxelization of the shoulder joint quaternions. (b) Extracted implicit surface. (c) Voxelization of the elbow joint Euler angles. (c) Extracted flat implicit surface.

As shown in Fig. 2(a,c), the resulting voxel arrays already represents the shape. We then simply place a spherical primitive at the center of each voxel and take its radius of influence to be half the width of the voxel, yielding the implicit surfaces depicted by Fig. 2(b,d), where $iso = 7.0$ and the stiffness $k = 20.0$.



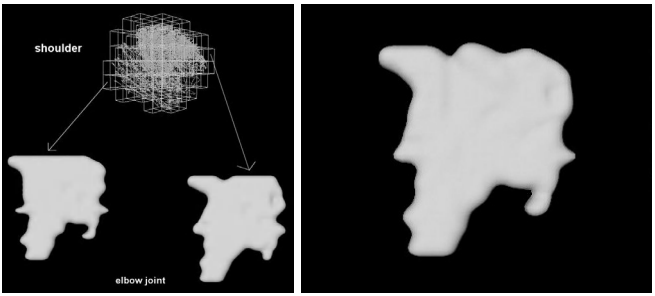
(a) (b)

**Fig. 3.** Comparing subjects against each other. In black, the data for the female reference subject we used to compute the field function $F$ of Eq. 2. In gray, the data corresponding to a second female subject (a) and to a male subject (b). We computed the average distance in terms of closest points between each cloud set, as well as the standard deviation. For (a), this yields an average distance of 0.0403 and a standard deviation of 0.0500. For (b), we obtain an average distance of 0.0314 and a standard deviation of 0.0432.

To illustrate the relative insensitivity of these measurements across subjects, we have gathered motion data for two additional people, one of each sex. In Fig. 3, we overlay the sets of quaternions for each additional person on those corresponding to the reference subject. Visual inspection in 3–D shows that they superpose well. This is confirmed by computing the average closest-point distance between the points of the three data sets, as well as the corresponding standard deviation.

### 4.2 Representing Dependencies

The method described above treats the data for the shoulder and the elbow independently, which does not account for known anatomical dependencies. To remedy this, we now take advantage of the voxel structure as follows. Each voxel for the parent joint, the shoulder in our case, defines a local cluster of similar joint positions, which we will refer to as *keyframe voxels*. As shown in Fig. 4(a), for each one, we compute the implicit *keyframe surface* corresponding to the subset of child joint rotations that have been observed for those positions of the parent joint. This follows directly from the fact that we simultaneously measure shoulder *and* elbow rotations. As shown in Fig. 4(b), to refine this representation and ensure a smoother transition between child joint limits from one keyframe voxel to the next, we can compute intermediate keframe surfaces by morphing between neighboring ones [29].




(a) (b)

**Fig. 4.** Hierarchical joint limits. (a) Two keyframe voxels and the corresponding keyframe surfaces. (b) Intermediate keyframe surface obtained midway through morphing one keyframe surface into the other.

## 5 Enforcing Constraints during Tracking

To validate our approach to enforcing joint limits, we show that it dramatically increases the performance of an earlier system [12] that fits body models to stereo-data acquired using synchronized video cameras. It relies on attaching implicit surfaces, also known as soft objects, to an articulated skeleton to represent body shape. The field function of the primitives however differs from the one used for defining our joint limits in the sense that its density field is exponential, which increase the robustness of the system in the presence of erroneous data points. The skin is taken to be a level set of the sum of these fields. Defining the body model surface in this manner yields an algebraic distance function from 3–D points to the model that is differentiable. We can therefore formulate the problem of fitting our model to the stereo data in each frame as one of minimizing the sum of the squares of the distances of the model to the cloud of points produced by the stereo.

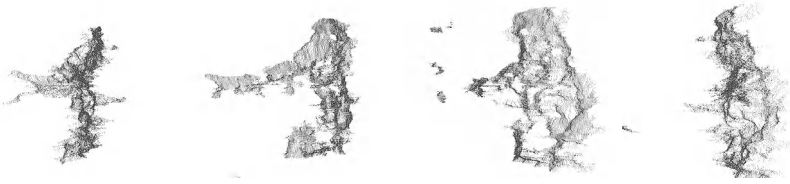


**Fig. 5.** Stereo data for a subject standing in the capture volume, rotated from a left-side view to a right-side view.

The stereo data is depicted by Fig. 5. It was acquired using a Digiclops$^{tm}$ operating at a $640 \times 480$ resolution and a 14Hz framerate. It is very noisy, lacks depth, and gives no information on the side or the back of the subject. As a result, in the absence of constraints, there are many sets of motion parameters that fit the data almost as well, most of which correspond to anatomically impossible postures.

In this Section, we will show that enforcing the constraints using the formalism allows to eliminate these impossible postures very effectively and results in much more robust tracking.

### 5.1 Constrained Least Squares

In the absence of constraints, the posture in each frame is obtained by least-squares minimization of the algebraic distance of the stereo data points to the skin surface, defined as a level set of the field function of Eq. 2. Enforcing hierarchical constraints can be effectively achieved using well known task-priority strategies. Here we use a damped least-squares method that can handle potentially conflicting constraints [11]: When a high-priority constraint is violated, the algorithm projects the invalid posture onto the closest valid one, which requires computing the pseudo-inverse of its Jacobian matrix with respect to state variables, which in our case are the rotational values of the model's joints. When a lower-priority constraint is violated, the algorithm reprojects the Jacobians into the null-space of the higher level constraints so that enforcing the lower-order constraint does not perturb the higher level one.

More specifically, the body is modeled as an articulated structure to which we we attach a number of volumetric primitives. This lets us define a field function $D(x, \Theta)$ that represents the algebraic distance of a 3–D point $x$ to the skin surface given the vector of joint angles values $\Theta = [\Theta_1, ..., \Theta_m]$. In the absence of constraints, fitting the model to a set of $n$ 3–D points $x_{i,1 \leq i \leq n}$ simply amounts to minimizing

$$\sum_{i=1}^{n} D(x_i, \Theta)^2 \quad ,$$

with respect to $\Theta$. Given the Jacobian matrix $J_D = (\frac{\partial D(x_i, \Theta)}{\partial \Theta_j})_{1 \leq i \leq n, 1 \leq j \leq m}$ and its pseudo-inverse $J_D^+$, this involves iteratively adding to $\Theta$ increments proportional to

$$\Delta\Theta_0 = J_D^+[D(x_1, \Theta), ..., D(x_n, \Theta)]^t \ .$$

Let us assume we are given a vector of constraints $C$ with Jacobian matrix $J_C$. The problem becomes minimizing $D$ subject to $C(\Theta) = 0.0$. This can be done very much in the same way as before, except that the increments are now proportional to

$$J_C^+ C(\Theta) + (I - J_C^+ J_C)\Delta\Theta_0 \ ,$$

where $(I - J_C^+ J_C)$ is the projector into the null space of $C$. This extends naturally to additional constraints with higher levels of priority, but additional care must be taken when constructing the projectors [11].

In short, all that is needed to enforce the constraints, is the ability to compute their Jacobian with respect to state variables. The implicit surface formulation of Section 4 lets us do this very simply:

1. For the parent joint, determine whether its rotation is valid by evaluating the function $F$ of Eq. 2 and its derivatives with respect to joint angles if not. In other words, the higher priority constraint can be expressed as $max(0, iso - F(\Theta))$ or, equivalently, treated as an inequality constraint.
2. For the child joint, determine to which voxel its parent rotation belongs, load the corresponding child joint limits, and verify its validity and, in case of an invalid rotation, evaluate the derivatives using the corresponding implicit surface representation. This allows us to express a lower priority constraint using the corresponding field function.

This results in an algorithm that fits the model to data, while enforcing the joint angles constraints at a minimal additional computational cost.

### 5.2 Tracking Results

We applied unconstrained and constrained tracking to several 100-frame long sequences, which corresponds to a little over 7 seconds at 14 Hz. In each sequence, the subject moves and rotates her right arm and elbow. In Figs 6, 7, and 8, we reproject the recovered 3–D skeleton onto one of the images. We also depict the skeleton as seen from a slightly different view to show whether or not the recovered position is feasible or not.

The unconstrained tracker performs adequately in many cases, but here we focus on the places where it failed, typically by producing the solution that matches the data but is not humanly possible. Among other things, this can be caused by the sparsity of the data or by the fact that multiple state vectors can yield limb positions that are almost indistinguishable given the poor quality of the data. We show that enforcing hierarchical joint limits on the shoulder and elbow joints during tracking allows our system to overcome these problems.

The interested reader can download mpeg movies for Figs. 7 and 8 from our http://cvlab.epfl.ch/research/body/limits/limits.html web site. They include the complete sequences along with depictions of the fit of the model to the 3–D data that are easier to interpret than the still pictures that appear in the printed version of the paper.

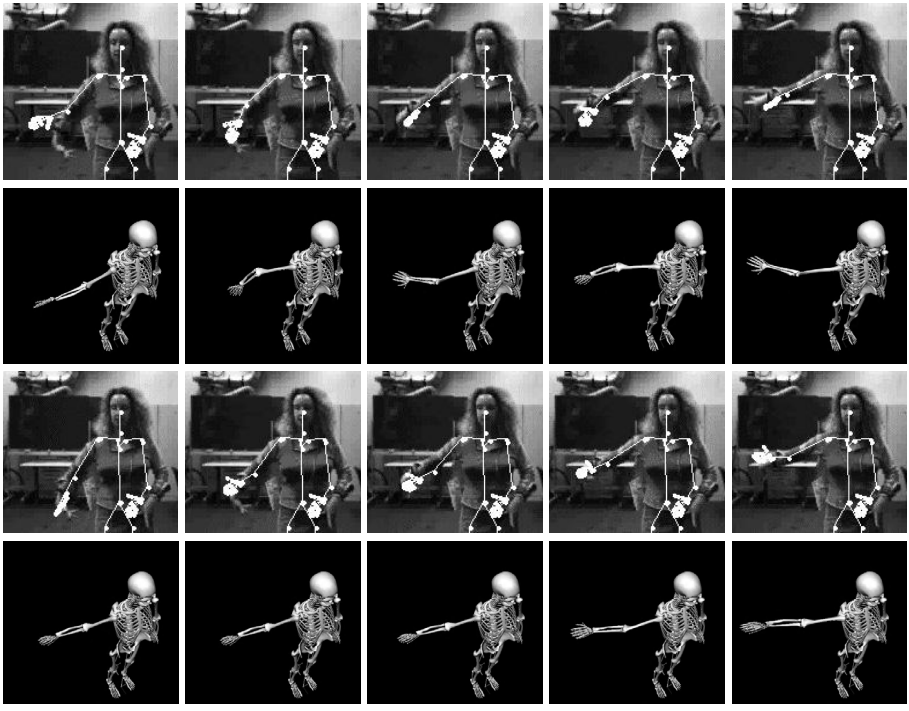



**Fig. 6.** Top rows: Unconstrained tracking. Bottom rows: Tracking with joint limits enforced. Up until the first frame shown here, the arm is tracked correctly in both cases. However, at frame 42, an elbow flexion occurs. In the unconstrained case, this is accounted for by backward bending of the elbow joint, which results in the correct reprojection but the absolutely impossible position of frame 56. By contrast, when the constraints enforced, the reprojection is just as good but the position is now natural.

## 6 Conclusion

We have proposed a implicit surface based approach to representing joint limits that account for both intra- and inter-joint dependencies. We have developed a protocol for instantiating this representation from motion capture data and shown that it can be effectively used to improve the performance of a body-tracking algorithm.

This effectiveness largely stems from the fact that our implicit surface representation allows us to quickly evaluate whether or not a constraint is violated and, if required, to enforce it using standard constrained optimization algorithms. We have demonstrated this in the specific case of the shoulder and elbow but the approach is generic and could be transposed to other joints, such as the hip and knee or the many coupled articulations in the hands and fingers.

The quality of the data we use to create our representation is key to its accuracy. The current acquisition process relies on optical motion capture. It is reasonably simple and fast, but could be improved further: Currently, when

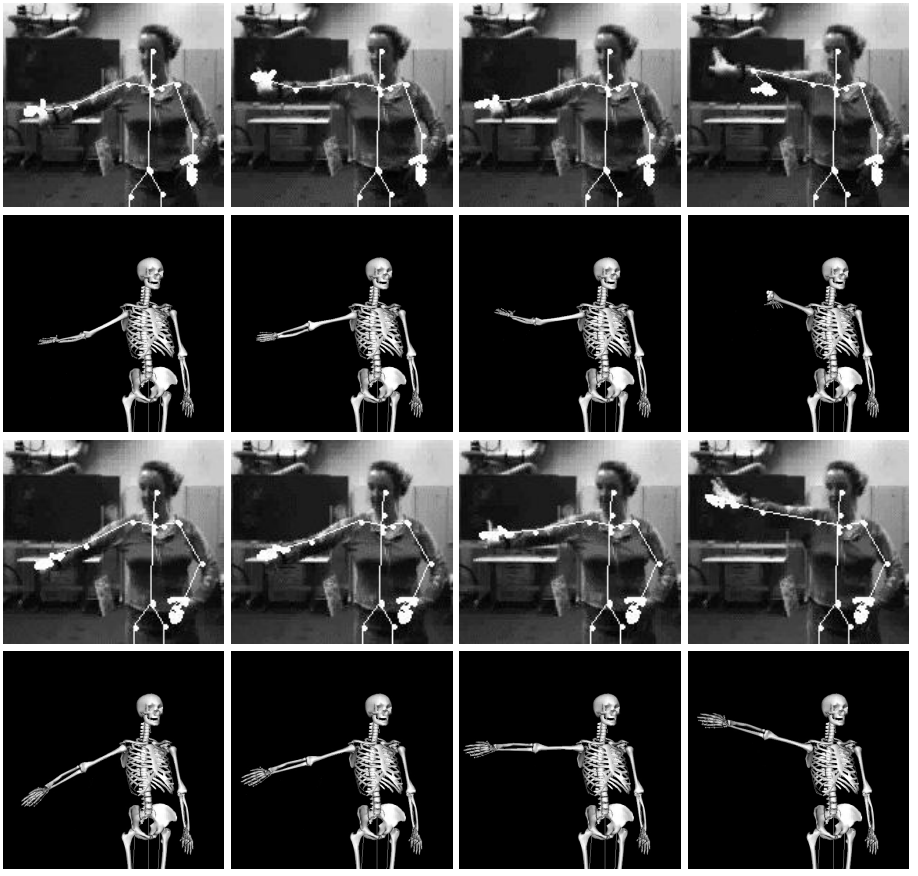



**Fig. 7.** Top rows: Unconstrained tracking. Bottom rows: Tracking with joint limits enforced. Tracking without constraints results in excessive shoulder axial rotation at frame 50, followed by wildly invalid elbow extension on top of the incorrect shoulder twisting at frame 51. In this frame, there happens to be very little data for the forearm, which ends up being erroneously "attracted" by the data corresponding to the upper arm. As can be seen in the bottom rows, these problems go away when the constraints are enforced. The corresponding mpeg movies are available on the web at http://cvlab.epfl.ch/research/body/limits/limits.html#Capture.

sampling the range of motion of a joint, we have no immediate feed-back on whether we have effectively sampled the entire attainable space. To remedy this problem, we will consider designing an application that provides immediate visual feed-back directly during motion acquisition. This should prove very useful when extending the proposed technique to larger hierarchies of joints than the parent-and-child one considered in this paper.

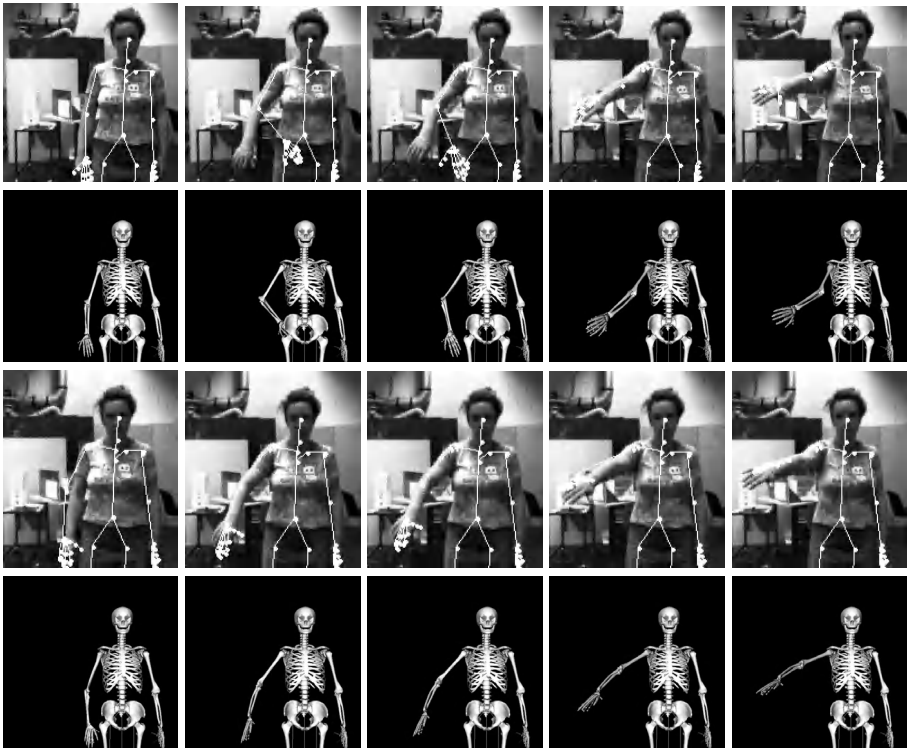



**Fig. 8.** Top rows: Unconstrained tracking. Bottom rows: Tracking with joint limits enforced. In the absence of constraints, the shoulder axial rotation is wrong from frame 1 onwards. In frames 23 to 25, this results in the arm being erroneously "attracted" by the 3–D data corresponding to the hip. The tracker then recovers in frame 31, only to yield an invalid elbow flexion in frame 34. Once again, these problems disappear when the constraints are enforced. The corresponding mpeg movies are available on the web at http://cvlab.epfl.ch/research/body/limits/limits.html#Capture.

## References

1. Gavrila, D.: The Visual Analysis of Human Movement: A Survey. Computer Vision and Image Understanding **73** (1999)
2. Moeslund, T., Granum, E.: Pose estimation of a human arm using kinematic constraints. In: Scandinavian Conference on Image Analysis, Bergen, Norway (2001)
3. Moeslund, T.: Computer Vision-Based Motion Capture of Body Language. PhD thesis, Aalborg University, Aalborg, Denmark (2003)
4. Rehg, J.M., Morris, D.D., Kanade, T.: Ambiguities in Visual Tracking of Articulated Objects using 2–D and 3–D Models. International Journal of Robotics Research **22** (2003) 393–418
5. Bregler, C., Malik, J.: Tracking People with Twists and Exponential Maps. In: Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA (1998)
6. Demirdjian, D.: Enforcing constraints for human body tracking. In: Workshop on Multi-Object Tracking. (2003)

7. Sminchisescu, C., Triggs, B.: Estimating articulated human motion with covariance scaled sampling. International Journal of Robotics Research (2003)
8. Engin, A., Tümer, S.: Three-dimensional kinematic modeling of the human shoulder complex. Journal of Biomechanical Engineering **111** (1989) 113–121
9. Herda, L., Urtasun, R., Hanson, A., Fua, P.: An automatic method for determining quaternion field boundaries for ball-and-socket joint limits. International Journal of Robotics Research **22** (2003) 419–436
10. Shoemake, K.: Animating Rotation with Quaternion Curves. Computer Graphics, SIGGRAPH Proceedings **19** (1985) 245–254
11. Baerlocher, P., Boulic, R.: An Inverse Kinematics Architecture for Enforcing an Arbitrary Number of Strict Priority Levels. The Visual Computer (2004)
12. Plänkers, R., Fua, P.: Articulated Soft Objects for Multi-View Shape and Motion Capture. IEEE Transactions on Pattern Analysis and Machine Intelligence (2003)
13. Hatze, H.: A three-dimensional multivariate model of passive human joint torques and articular boundaries. Clinical Biomechanics **12** (1997) 128–135
14. Kodek, T., Munich, M.: Identifying Shoulder and Elbow Passive Moments and Muscle Contributions. In: International Conference on Intelligent Robots and Systems. (2002)
15. Johnston, R., Smidt, G.: Measurement of hip joint motion during walking. Journal of Bone and Joint Surgery **51** (1969) 1083–1094
16. Meskers, C., Vermeulen, H., de Groot, J., der Helm, F.V., Rozing, P.: 3d shoulder position measurements using a six-degree-of-freedom electromagnetic tracking device. Clinical Biomechanics **13** (1998) 280–292
17. der Helm, F.V.: A standardized protocol for motion recordings of the shoulder. In: Conference of the International Shoulder Group, Masstritcht, Netherlands (1997)
18. Bao, H., Willems, P.: On the kinematic modelling and the parameter estimation of the human shoulder. Journal of Biomechanics **32** (1999) 943–950
19. Maurel, W.: 3d modeling of the human upper limb including the biomechanics of joints, muscles and soft tissues (1998)
20. Wang, X., Maurin, M., Mazet, F., Maia, N.D.C., Voinot, K., Verriest, J., Fayet, M.: Three-dimensional modelling of the motion range of axial rotation of the upper arm. Journal of Biomechanics **31** (1998) 899–908
21. Bobick, N.: Rotating objects using quaternions. Game Developer **2, Issue 26** (1998)
22. Watt, A., Watt, M.: Advanced animation and rendering techniques (1992)
23. Grassia, F.: Practical parameterization of rotations using the exponential map. Journal of Graphics Tools **3** (1998) 29–48
24. Pervin, E., Webb, J.: Quaternions for computer vision and robotics. In: Conference on Computer Vision and Pattern Recognition, Washington, D.C. (1983) 382–383
25. Faugeras, O.: Three-Dimensional Computer Vision: a Geometric Viewpoint. MIT Press (1993)
26. Bloomenthal, J.: Calculation of reference frames along a space curve. In Glassner, A., ed.: Graphics Gems. Academic Press, Cambridge, MA (1990) 567–571
27. Tsingos, N., Bittar, E., Gascuel, M.: Implicit surfaces for semi-automatic medical organs reconstruction. In: Computer Graphics International, Leeds, UK (1995) 3–15
28. Lorensen, W., Cline, H.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. In: Computer Graphics, SIGGRAPH Proceedings. Volume 21. (1987) 163–169
29. V., R., A., F.: Shape transformations using union of spheres. Technical Report TR-95-30, Department of Computer Science, University of British Columbia (1995)

# Separating Specular, Diffuse, and Subsurface Scattering Reflectances from Photometric Images*

Tai-Pang Wu and Chi-Keung Tang

Vision and Graphics Group
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong

**Abstract.** While subsurface scattering is common in many real objects, almost all separation algorithms focus on extracting specular and diffuse components from real images. In this paper, we propose an appearance-based approach to separate non-directional subsurface scattering reflectance from photometric images, in addition to the separation of the off-specular and non-Lambertian diffuse components. Our mathematical model sufficiently accounts for the photometric response due to non-directional subsurface scattering, and allows for a practical image acquisition system to capture its contribution. Relighting the scene is possible by employing the separated reflectances. We argue that it is sometimes necessary to separate subsurface scattering component, which is essential to highlight removal, when the object reflectance cannot be modeled by specular and diffuse components alone.

## 1 Introduction

Photometric appearance of an object depends on surface geometry, material property, illumination and viewing direction of the camera. The ubiquitous Lambertian assumption used in computer vision algorithms is seldom satisfied, making them error prone to specular highlight, off-specular reflection, and subsurface scattering. Almost all reflectance separation algorithms separate specular and Lambertian diffuse reflectances. Lin and Lee [8] made use of the Lafortune model [5] to separate off-specular and non-Lambertian diffuse components. The Lafortune model assumes the bidirectional reflectance distribution function (BRDF), which describes the ratio of outgoing radiance to incoming irradiance at the *same* surface point locally.

However, many real objects such as wax, paper, and objects with translucent protective coating do not belong to this category. Subsurface scattering is common, where the outgoing radiance observed at a surface point may be due to the incoming irradiances at *different* surface points. Fig. 1 illustrates a common phenomenon. To account for subsurface scattering, the more general bi-directional surface scattering reflectance distribution function (BSSRDF) should be used [4]. However, BSSRDF is very difficult to capture without knowledge of object geometry and material property.

In this paper, we propose an appearance-based method, and derive the mathematical model for reflectance separation that accounts for non-directional subsurface scattering,

* This work is supported by the Research Grant Council of Hong Kong Special Administration Region, China: HKUST6193/02E.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 419–433, 2004.
© Springer-Verlag Berlin Heidelberg 2004

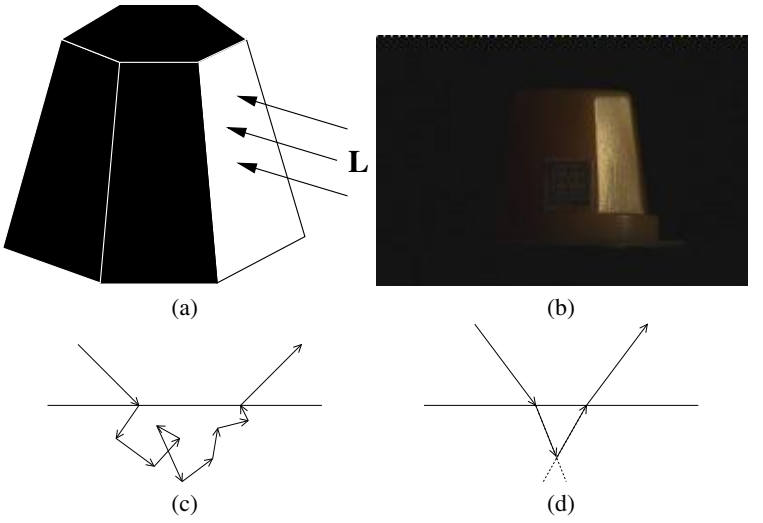



**Fig. 1.** Ideally, only the face visible to the directional light **L** can be seen (a), if the object reflectance is explained by BRDF. In reality, due to subsurface scattering, significant radiance is observed on other faces (b). In this paper, we model non-directional subsurface scattering (c), but not single scattering (d) where a dominant outgoing radiance direction exists.

Fig. 1. We do not model single scattering [2], which has a dominant direction in the outgoing radiance. Given images only, we decompose the input into three photometric components: *off-specular* $I_s$, non-Lambertian or *local diffuse* $D$, and *non-directional subsurface scattering* $I_{scat}$. BSSRDF can describe $I_s$, $D$ and $I_{scat}$, but BRDF can only describe the local $I_s$ and $D$. Our approach avoids the difficult problem of capturing/recovering the BRSSDF. Yet, it is versatile enough to describe a wide range of lighting conditions, re-light the scene and remove specular highlight. Transparency and single scattering are the topics of our ongoing research.

## 2 Related Work

We review some representative reflectance models used in computer vision and computer graphics. In computer vision, given images, reflectance models are often used in parameter fitting for seperating specular and diffuse components. In computer graphics, given parameters to the model, photorealistic images are rendered to simulate specular and diffuse effects.

**Forward approach: graphics rendering.** Cook and Torrance [1] introduced a reflectance model to describe the distribution of the reflected light and color shift as the reflectance changes with incidence angle. Later, Poulin and Fournier [10] introduced another model to handle anisotropic reflection. This model considers the micro-structure of material surface as small cylinders. Oren and Nayar [9] proposed a model to describe complex

geometry and radiometric phenomena to account for non-Lambertian diffuse reflection. They assume or approximate light transport by bi-directional reflection distribution function (BRDF). To account for reflectance due to subsurface scattering, Hanrahan and Krueger [2] introduced a model that considers the subsurface reflectance due to back-scattering in a layered medium, by using transport theory. However, this model cannot capture the diffusion of light across shadow boundaries. Jensen et al. [4] introduced another model to overcome this problem. This model approximates the bi-directional surface scattering reflectance distribution function (BSSRDF), which does not restrict the light ray to enter and exit the surface at the same location. In fact, BRDF is the special case of BSSRDF.

Although the rendering method by Jensen et al. [4] predicts the appearance of translucent materials very well, complete knowledge on the scene (the geometry of the scene, object materials, lighting condition and camera configuration) should be known. Shape-from-shading method may be used to estimate the surface geometry information. For example, Ragheb and Hancock [11] used an iterated conditional modes algorithm to estimate the Lambertian and specular reflectance components and recovering local surface normals. Hertzmann and Seitz [3] introduced a method to recover the local surface normals with a reference object whose geometry is known. However, current methods use BRDF, but not the more general BSSRDF, which is capable of explaining subsurface scattering.

**Backward approach: reflectance separation.** When accurate geometry and reflectance information are not available, image-based methods are used. The camera position is usually fixed. Shashua [12] proved that, if Lambertian diffusion is assumed, images illuminated under any lighting direction can be produced by linear combination of three photometric images captured under three linearly independent lighting vectors.

Lin and Lee [7] derived an approximate specular reflectance model based on the Torrance-Sparrow model, where the local specular effect can be expressed by logarithms of three intensity-normalized photometric images under certain illumination and surface conditions. In [6], they introduced a method for representing diffuse and specular reflections for objects of uniform surface roughness using four photometric images. To relight the image, only four images are needed without the knowledge of surface geometry. By extending the same idea to the Lafortune model [5], Lin and Lee [8] introduced a method capable of separating any number of reflectances into distinct, manageable components such that the model parameters can be estimated readily. Any novel lighting condition can be simulated to relight images, by non-linear combination of the photometric images of the separated reflectances and the model parameters.

**Our contribution.** We show that by transforming light vectors in the set of equations, Lin and Lee's method can be generalized to include a non-directional subsurface scattering term $I_{scat}$ in the diffuse term. By ensuring invariance in the resulting appearance and preserving outgoing radiance energy, the proposed transformation of light vectors is legitimate.

In implementation, our key mathematical derivation is translated into the estimation of point spread functions (PSF) to estimate the $I_{scat}$ term. A new and practical image

acquisition system is built to capture subsurface scattering reflectance from real objects of unknown geometry.

Potential application is the removal of the specular highlight term $I_s$ in the presence of subsurface scattering $I_{scat}$ (e.g., paper, wax). Both are problematic to many computer vision algorithms. Alternatively, $I_s$, $D$, and $I_{scat}$ can be combined non-linearly to simulate a novel image under a different lighting condition.

The rest of our paper is organized as follows. Section 3 discusses different choices of reflectance models. In section 4, we derive our new reflectance representation for photometric images. Section 5 discusses how the parameters of our reflectance representation can be estimated. The new image acquisition method is introduced in Section 6. Section 7 presents the implementation and shows our results. We conclude our work in Section 8.

## 3 Reflectance Models

Except for mirror-like surface material such as metal, most material exhibits certain extent of translucence which is resulted by subsurface scattering. This phenomenon is described by BSSRDF. Mathematically, BSSRDF $S$ is defined as:

$$dL_o(x_o, \overrightarrow{\omega_o}) = S(x_i, \overrightarrow{\omega_i}; x_o, \overrightarrow{\omega_o}) d\Phi_i(x_i, \overrightarrow{\omega_i}) \tag{1}$$

where $L_o(x_o, \overrightarrow{\omega_o})$ is the outgoing radiance at point $x_o$ in direction $\overrightarrow{\omega_o}$, $\Phi_i(x_i, \overrightarrow{\omega_i})$ is the incident flux at the point $x_i$ from direction $\omega_i$. When $i = o$, $S$ becomes BRDF. By integrating the incident irradiance over all incoming directions and area $A$ on the surface, the total outgoing radiance at $x_o$ is:

$$L_o(x_o, \overrightarrow{\omega_o}) = \int_A \int_{2\pi} S(x_i, \overrightarrow{\omega_i}; x_o, \overrightarrow{\omega_o}) L_i(x_i, \overrightarrow{\omega_i}) (\overrightarrow{n_i} \cdot \overrightarrow{\omega_i}) d\overrightarrow{\omega_i} dA(x_i) \tag{2}$$

Actually, $L_i(x_i, \overrightarrow{\omega_i})(\overrightarrow{n_i} \cdot \overrightarrow{\omega_i}) d\overrightarrow{\omega_i}$ is the energy of the incident flux defined by Cook and Torrance [1]. In other words, this equation describes how the incident flux (energy) is distributed from location $x_i$ to $x_o$. With this equation, a good prediction on the appearance of translucent objects is possible. However, $S$ should be known *a priori* for all $x$ and $\omega$. Recovering the BSSRDF $S$ is difficult without special equipment. It is unknown if a gonioreflectometer (used to capture BRDF) can be used to measure BSSRDF.

An alternative is to approximate $S$ by fitting analytical functions. However, almost all reflectance models approximate BRDF, except a few on BRSSDF [4]. Our appearance-based model is capable of representing a class of BRSSDF, by generalizing the Lafortune model [5]. Lin and Lee [8] investigated the Lafortune model, which considers each type of reflectance as a non-linear parametric primitive function. The final object appearance is the linear combination of these primitives. This model provides high flexibility. The other possible choice is the Cook and Torrance model [1]. This model gives a good prediction on specular effect and color shift at grazing angle. Although Lin and Lee [6] showed that the parameters of Cook and Torrance model can be estimated under limited viewing conditions, it is still too complex to use. As a result, we chose the Lafortune model. The Lafortune model [5] is defined as

$$R(\mathbf{L}, \mathbf{V}) = \sum_i [C_{x,i} L_x V_x + C_{y,i} L_y V_y + C_{z,i} L_z V_z]^{n_i} \tag{3}$$

where $i$ is the index of primitives, $\mathbf{L} = (L_x, L_y, L_z)^T$ is light direction, $\mathbf{V} = (V_x, V_y, V_z)^T$ is viewing direction, $C$ is weighting coefficient and $x, y$ and $z$ index a local coordinate system with the $z$-axis aligned with the surface normal, and the $x$-axis and $y$-axis aligned with the principal directions of anisotropy [5], except for unusual type of anisotropy. According to [8], since the $z$-axis is aligned with a surface normal, we have

$$L_z V_z = (\mathbf{N} \cdot \mathbf{L})(\mathbf{N} \cdot \mathbf{V}) \tag{4}$$

$$L_x V_x + L_y V_y = (\mathbf{L} \cdot \mathbf{V}) - (\mathbf{N} \cdot \mathbf{L})(\mathbf{N} \cdot \mathbf{V}) \tag{5}$$

With this relationship, Lin and Lee [8] showed that the non-Lambertian diffuse and off-specular components, $I_d(x)$ and $I_s(x)$, at pixel $x$ on image $I$ can be expressed as:

$$I_d(x) = \rho(x) \frac{n_d + 2}{2\pi} [(\mathbf{N}(x) \cdot \mathbf{L})(\mathbf{N}(x) \cdot \mathbf{V})]^{n_d} \tag{6}$$

$$I_s(x) = [C_1(\mathbf{L} \cdot \mathbf{V}) + C_2(\mathbf{N}(x) \cdot \mathbf{L})(\mathbf{N}(x) \cdot \mathbf{V})]^{n_s} \tag{7}$$

where $\rho(x)$ is surface albedo, $\mathbf{N}(x)$ is normal at $x$, $n_d$ and $n_s$ are the exponents for non-Lambertian diffuse and off-specular components respectively, $C_1 = C_x$ and $C_2 = C_z - C_x$.

In [1], it is stated that the specular component is the result of reflection from the surface of material, and the diffuse component is that of internal scattering or multiple surface reflections. Internal scattering is due to the penetration of incident light beneath the material surface. With this definition, we embed the effect of subsurface scattering into the diffuse term in the equations.

## 4 Reflectance Representation

We assume directional light source in our derivation. The following notations are useful in the discrete formulation for our implementation: a small surface patch on a 3D object is represented by $\overrightarrow{v}$. $x$ is the (quantized) image point of $\overrightarrow{v}$. $\mathbf{L}$ as the directional lighting vector, which encapsulates both the direction and intensity, $\mathbf{N}$ is an overloaded function that returns the normal direction at $\overrightarrow{v}$, or the normal direction at $x$ ( $\mathbf{N}(\overrightarrow{v}) = \mathbf{N}(x)$ if the image of $\overrightarrow{v}$ is $x$). $m_{\overrightarrow{v_0}}(\overrightarrow{v_i})$ is the discrete representation for the BSSRDF $S$ between $\overrightarrow{v_0}$ and $\overrightarrow{v_i}$, which specifies how the energy of incoming ray distributes from $\overrightarrow{v_i}$ to $\overrightarrow{v_0}$.

Using the above notations, the discrete version of Eqn. (2) is re-written as:

$$energy(\overrightarrow{v_0}, \mathbf{L}) = \sum_{i=0}^{n} m_{\overrightarrow{v_0}}(\overrightarrow{v_i}) \mathbf{N}(\overrightarrow{v_i}) \cdot \mathbf{L} \tag{8}$$

Without loss of generality, $energy(\overrightarrow{v_0}, \mathbf{L})$ indicates the total outgoing radiance at $\overrightarrow{v_0}$, $n$ is the total number of surface patch on the surface, and $i$ is the index of the subsurface patch. It is important to note that, for $m$, the only input parameter is a surface patch $\overrightarrow{v}$, which is different from the BSSRDF $S$ described in section 3 requiring both the position and direction of incident and outgoing rays. The main reason is that we do not model the effect of single scattering, which is our future work. It requires the knowledge of light ray directions.

From Eqn. (8), we can consider that $m$ is a PSF which models the diffusion of light due to *local diffuse* $D$ and *subsurface scattering* $I_{scat}$. When a light ray enters a highly scattering medium from an arbitrary direction at $\overrightarrow{v}$, the light distribution is non-directional, since the light is scattered and absorbed randomly inside the medium. Since the PSF depends on surface geometry and material property of the object, the PSF at each $\overrightarrow{v}$ is different in general. Therefore, the subscript of $m_{\overrightarrow{v_0}}(\overrightarrow{v_i})$ reflects that it is a local variable, and surface patch $\overrightarrow{v_i}$ is the only input parameter.

By rotating the local coordinate system of each surface patch $\overrightarrow{v_i}$ with rotation matrix $\psi_i$ such that $\mathbf{N}(\overrightarrow{v_i})$ aligns with $\mathbf{N}(\overrightarrow{v_0})$, Eqn. (8) becomes:

$$\begin{aligned} energy(\overrightarrow{v_0}, \mathbf{L}) &= \sum_{i=0}^{n} m_{\overrightarrow{v_0}}(\overrightarrow{v_i})\mathbf{N}(\overrightarrow{v_0}) \cdot (\psi_i \mathbf{L}) \\ &= \mathbf{N}(\overrightarrow{v_0}) \cdot [\sum_{i=0}^{n} m_{\overrightarrow{v_0}}(\overrightarrow{v_i})(\psi_i \mathbf{L})] \qquad (9) \\ &= \mathbf{N}(\overrightarrow{v_0}) \cdot \mathbf{L}_{final} \qquad (10) \end{aligned}$$

where $\mathbf{L}_{final} = \sum_{i=0}^{n} m_{\overrightarrow{v_0}}(\overrightarrow{v_i})(\psi_i \mathbf{L})$. From Eqn. (10), the total outgoing radiance at $\overrightarrow{v_0}$ can be considered as resizing and jittering of the original $\mathbf{L}$ to produce $\mathbf{L}_{final}$. By putting Eqn. (10) into (6), we express $I_d(x)$ as:

$$\begin{aligned} &\rho(x)\frac{n_d+2}{2\pi}[(\mathbf{N}(\overrightarrow{v_0}) \cdot \mathbf{L}_{final})(\mathbf{N}(x) \cdot \mathbf{V})]^{n_d} \\ &= \rho(x)\frac{n_d+2}{2\pi}[(\mathbf{N}(x) \cdot \mathbf{L}_{final})(\mathbf{N}(x) \cdot \mathbf{V})]^{n_d} \\ &= \rho(x)\frac{n_d+2}{2\pi}\{[m_{\overrightarrow{v_0}}(\overrightarrow{v_0})\mathbf{N}(x) \cdot (\psi_0 \mathbf{L}) + \mathbf{N}(x) \cdot [\sum_{i=1}^{n} m_{\overrightarrow{v_0}}(\overrightarrow{v_i})(\psi_i \mathbf{L})]][\mathbf{N}(x) \cdot \mathbf{V}]\}^{n_d} \qquad (11) \\ &= \rho(x)\frac{n_d+2}{2\pi}\{[m_{\overrightarrow{v_0}}(\overrightarrow{v_0})\mathbf{N}(x) \cdot \mathbf{L} + \mathbf{N}(x) \cdot \mathbf{L}_{scat}][\mathbf{N}(x) \cdot \mathbf{V}]\}^{n_d} \qquad (12) \end{aligned}$$

where $\psi_0$ is defined to be an identity matrix, and $\mathbf{L}_{scat} = \sum_{i=1}^{n} m_{\overrightarrow{v_0}}(\overrightarrow{v_i})(\psi_i \mathbf{L})$. Then, the $n_d$-th root of Eqn. (12) is

$$\begin{aligned} I_d(x)^{\frac{1}{n_d}} &= [\rho(x)\frac{n_d+2}{2\pi}]^{\frac{1}{n_d}}[m_{\overrightarrow{v_0}}(\overrightarrow{v_0})\mathbf{N}(x) \cdot \mathbf{L}][\mathbf{N}(x) \cdot \mathbf{V}] \\ &\quad + [\rho(x)\frac{n_d+2}{2\pi}]^{\frac{1}{n_d}}[\mathbf{N}(x) \cdot \mathbf{L}_{scat}][\mathbf{N}(x) \cdot \mathbf{V}] \\ &= m_{\overrightarrow{v_0}}(\overrightarrow{v_0})I_{local}(x)^{\frac{1}{n_d}} + I_{scat}(x)^{\frac{1}{n_d}} \qquad (13) \end{aligned}$$

In Eqn. (13), $I_{local}$ and $I_{scat}$ have the same form as Eqn. (6), which contribute to the diffuse component of the outgoing radiance $I_d$. $I_{local}$ accounts for the outgoing radiance locally reflected at $\overrightarrow{v_i}$ without any subsurface scattering (since $\psi_0$ is an identity matrix). $I_{scat}$ accounts for the outgoing radiance resulting from incident irradiances at different patches. Using the PSF $m$, it can be thought that appropriate portions of $\mathbf{L}$ are "distributed" to $I_{local}$ and $I_{scat}$. Putting together, a captured image $I(x) = I_d(x) + I_s(x)$ is given by:

$$I(x) = [m_{\overrightarrow{v_0}}(\overrightarrow{v_0})I_{local}(x)^{\frac{1}{n_d}} + I_{scat}(x)^{\frac{1}{n_d}}]^{n_d} + I_s(x) \qquad (14)$$

Since any lighting vector $\mathbf{L}$ could be written as the linear combination of three linear independent lighting vectors $\mathbf{L_1}, \mathbf{L_2}$ and $\mathbf{L_3}$ such that

$$\mathbf{L} = \alpha_1 \mathbf{L_1} + \alpha_2 \mathbf{L_2} + \alpha_3 \mathbf{L_3}, \tag{15}$$

it is possible to represent the $n_d$-th root of the diffuse component as expressed in Eqn. (13) by combining three photometric images that are illuminated by $\mathbf{L_1}, \mathbf{L_2}, \mathbf{L_3}$. By putting Eqn. (15) into (11) and taking the $n_d$-th root in both sides,

$$\begin{aligned}
I_d(x)^{\frac{1}{n_d}} &= [\rho(x)\frac{n_d+2}{2\pi}]^{\frac{1}{n_d}} [m_{\overrightarrow{v_0}}(\overrightarrow{v_0})\mathbf{N}(x)\cdot(\alpha_1\mathbf{L_1}+\alpha_2\mathbf{L_2}+\alpha_3\mathbf{L_3})][\mathbf{N}(x)\cdot\mathbf{V}] \\
&+ [\rho(x)\frac{n_d+2}{2\pi}]^{\frac{1}{n_d}} [\mathbf{N}(x)\cdot[\sum_{i=1}^{n} m_{\overrightarrow{v_0}}(\overrightarrow{v_i})(\psi_i(\alpha_1\mathbf{L_1}+\alpha_2\mathbf{L_2}+\alpha_3\mathbf{L_3}))]][\mathbf{N}(x)\cdot\mathbf{V}] \\
&= m_{\overrightarrow{v_0}}(\overrightarrow{v_0})[\alpha_1 I_{local,1}(x)^{\frac{1}{n_d}} + \alpha_2 I_{local,2}(x)^{\frac{1}{n_d}} + \alpha_3 I_{local,3}(x)^{\frac{1}{n_d}}] \\
&+ [\rho(x)\frac{n_d+2}{2\pi}]^{\frac{1}{n_d}} [\mathbf{N}(x)\cdot(\alpha_1\mathbf{L}_{scat,1}+\alpha_2\mathbf{L}_{scat,2}+\alpha_3\mathbf{L}_{scat,3})][\mathbf{N}(x)\cdot\mathbf{V}] \\
&= m_{\overrightarrow{v_0}}(\overrightarrow{v_0})[\alpha_1 I_{local,1}(x)^{\frac{1}{n_d}} + \alpha_2 I_{local,2}(x)^{\frac{1}{n_d}} + \alpha_3 I_{local,3}(x)^{\frac{1}{n_d}}] \\
&+ [\alpha_1 I_{scat,1}(x)^{\frac{1}{n_d}} + \alpha_2 I_{scat,2}(x)^{\frac{1}{n_d}} + \alpha_3 I_{scat,3}(x)^{\frac{1}{n_d}}]
\end{aligned} \tag{16}$$

Similarly, by putting Eqn. (15) into (7) and taking the $n_s$-th root, the specular component can be represented by:

$$I_s(x)^{\frac{1}{n_s}} = \alpha_1 I_{s,1}(x)^{\frac{1}{n_s}} + \alpha_2 I_{s,2}(x)^{\frac{1}{n_s}} + \alpha_3 I_{s,3}(x)^{\frac{1}{n_s}} \tag{17}$$

Eqns (16) and (17) show that any photometric image can be represented by combining three images captured at independent lighting directions $\mathbf{L_1}, \mathbf{L_2}$ and $\mathbf{L_3}$.

## 5 Issues in Parameter Estimation

After deriving our reflectance separation, next we need to estimate the associated parameters. The desired method is to estimate the parameters directly from images. To achieve this, the method suggested by Lin and Lee [8] is a possible choice. First, the diffuse and specular components are related. Second, six images, acquired with different lighting vectors, are divided into two sets with three images each. Images in one set are described by the combination of the other set. Finally, the parameters of their representation are estimated by optimizing an objective function. Here, we need to find the relationship between the specular and our diffuse component that includes a subsurface scattering term.

If an object does not exhibit any light scattering effect, the term $I_{scat}(x)$ is zero for all the pixels on an captured image. Then, our derivation will be the same as [8]. Thus, the formulation of Lin and Lee [8] is a special case of ours. From [8], the relationship between local diffuse $I_{local,k}$ and specular component $I_{s,k}$ is:

$$I_{s,k}(x)^{\frac{1}{n_d}} = a_k + b(x) I_{local,k}(x)^{\frac{1}{n_d}} \tag{18}$$

where $k = 1, 2, 3$ are the indices of images, $a_k = C_1 \mathbf{L_k}\cdot\mathbf{V}$ and $b(x) = C_2/[\rho(x)\frac{n_d+2}{2\pi}]^{\frac{1}{n_d}}$. With this relationship, the image $I_k$ acquired can be expressed as:

$$I_k(x) = [m_{\overrightarrow{v_0}}(\overrightarrow{v_0}) I_{local,k}(x)^{\frac{1}{n_d}} + I_{scat,k}(x)^{\frac{1}{n_d}}]^{n_d} + [a_k + b(x) I_{local,k}(x)^{\frac{1}{n_d}}]^{n_s} \quad (19)$$

In order to simplify Eqn. (19), we define $D_k(x) = m_{\overrightarrow{v_0}}(\overrightarrow{v_0})^{n_d} I_{local,k}(x)$ and $B(x) = C_2 / [m_{\overrightarrow{v_0}}(\overrightarrow{v_0})^{n_d} \rho(x) \frac{n_d+2}{2\pi}]^{\frac{1}{n_d}}$. Eqn. (19) becomes:

$$I_k(x) = [D_k(x)^{\frac{1}{n_d}} + I_{scat,k}(x)^{\frac{1}{n_d}}]^{n_d} + [a_k + B(x) D_k(x)^{\frac{1}{n_d}}]^{n_s} \quad (20)$$

Similarly, adding the powers of $n_d$ and $n_s$ of Eqns (16) and (17) respectively:

$$\begin{aligned} I_k(x) = & [(\alpha_{1,k} D_1(x)^{\frac{1}{n_d}} + \alpha_{2,k} D_2(x)^{\frac{1}{n_d}} + \alpha_{3,k} D_3(x)^{\frac{1}{n_d}}) \\ & + (\alpha_{1,k} I_{scat,1}(x)^{\frac{1}{n_d}} + \alpha_{2,k} I_{scat,2}(x)^{\frac{1}{n_d}} + \alpha_{3,k} I_{scat,3}(x)^{\frac{1}{n_d}})]^{n_d} \\ & + \{\alpha_{1,k}[a_1 + B(x) D_1(x)^{\frac{1}{n_d}}] \\ & + \alpha_{2,k}[a_2 + B(x) D_2(x)^{\frac{1}{n_d}}] \\ & + \alpha_{3,k}[a_3 + B(x) D_3(x)^{\frac{1}{n_d}}]\}^{n_s} \end{aligned} \quad (21)$$

From the above two equations, we can see that the parameters needed to be estimated consist of five global parameters $\{a_1, a_2, a_3, n_d, n_s\}$ and seven local parameters $\{B(x), D_1(x),\ D_2(x), D_3(x), I_{scat,1}(x), I_{scat,2}(x), I_{scat,3}\}$. By using the framework of Lin and Lee [8], we have to capture $n$ images that are illuminated by $n$ different lighting vectors. Three of them ($k = 1, 2, 3$) are represented by Eqn. (20), and the others ($k \geq 4$) are represented by Eqn (21). Since we have $7p + 5$ unknowns, where $p$ is the total number of pixels in image $I$, $n \geq 8$ images are needed.

The values $\alpha_{i,k}$ for each of the image $k \geq 4$ may be approximated from pixels whose intensity does not exceed a threshold by the method suggested in [12]. In order to estimate the global and local parameters, we minimize the following error function:

$$Err = \sum_x [\lambda_1 \sum_{k=1}^{3} e(k,x)^2 + \lambda_2 \sum_{k=4}^{n} e(k,x)^2] \quad (22)$$

where $e(k,x)$ is an error function for $I_k(x)$ and $\lambda_1 = 1$ and $\lambda_2 = 2$ are weighting coefficients (as in [8]). In our implementation, $e(k,x)$ is image difference. The function *Err* is then minimized by some optimization algorithm such as Levenberg-Marquardt algorithm. It is suggested that the global parameters are estimated first before the local parameters are estimated pixel-by-pixel in order to reduce the computation load. The former can be done by selecting $p \geq \lceil (n-7)p + 5 \rceil$ pixels from the image and perform optimization with Eqn. (22).

Although everything above seems to be fine so far, there are some reasons that the above framework cannot be used directly for solving our reflectance Eqns (20) and (21). One problem is that there are too many local parameters, leading to a lot of local minima. Besides, too many images ($n - 3$, where $n \geq 8$) are required for the approximation of $\alpha_{i,k}$. Both problems complicate the optimization process and make it less stable. Worst, the local variable $I_{scat,k}(x)$ is *independent* of $D_k(x)$ and $I_s(x)$ in Eqn. (20). Given a pixel $x$, during the optimization, if specular component does not exist, the value of diffuse term $I_d(x)$ can be "distributed" to $D_k(x)$ and $I_{scat,k}(x)$ arbitrarily to produce (local) minima.

To solve these problems, some parameters should preferably be estimated first. Fixing them before the optimization process will make it more stable. Suppose that the value of $I_{scat,k}(x)$, where $k = 1, 2, 3$, can be found before the optimization stage, we shall have five global and four local parameters to estimate, and the number of images required becomes $n \geq 6$. Then, the number of parameters and the number of required images are the same as in [8].

In the following section, a new image acquisition system is built to estimate $I_{scat,k}(x)$.

## 6 Image Acquisition for Estimating $I_{scat}$

Except at the end of this section, we drop the subscript $k$ since the same image acquisition method is used for estimating all $I_{scat,k}$. Recall from Eqn. (13) that the PSF term $m$ is included into the subsurface scattering term $I_{scat}$.

If there is a *single* light ray incident at a surface patch at $\overrightarrow{v_0}$, we can warp the PSF on the object surface centered at $\overrightarrow{v_0}$. Refer to Eqn. (20), the scattering term $I_{scat}$ is completely independent of the $D$ and $I_s$ terms. Therefore, given a single ray incident at $\overrightarrow{v_0}$, except at $\overrightarrow{v_0}$, the outgoing radiance at other surface patches $\overrightarrow{v_i}, i \neq 0$, is due to subsurface scattering *only*. For these patches at $\overrightarrow{v_i}$, the corresponding images $D$ and $I_s$ are zero. Therefore, if we could produce such a single light ray incident at each $\overrightarrow{v}$ on the object surface, it would be possible to recover $I_{scat}$. However, it is difficult to build high precision equipment capable of shooting a single ray of light. A light slab produced by a ray box used in optics experiments covers more than one surface patches, as shown in Fig. 2(a). Following, we show how we deploy a light slab to estimate $I_{scat}$.

**Analysis.** Consider $I_{scat}(x)$ in Eqn. (13). By expanding $\mathbf{L}_{scat}$, which is the linear combination of rotated and resized $\mathbf{L}$ from $n$ surface patches, the relationship between energy contribution from each of the surface patch and $I_{scat}(x)$ can be written as:

$$I_{scat}(x)^{\frac{1}{n_d}} = I_{v,1}(x)^{\frac{1}{n_d}} + I_{v,2}(x)^{\frac{1}{n_d}} + ... + I_{v,n}(x)^{\frac{1}{n_d}} \tag{23}$$

where each $I_{v,i}(x)$ represents the images of light diffusion resulting from surface patch $i$. It shows that $I_{scat}(x)^{1/n_d}$ is just the linear combination of $I_{v,i}(x)^{1/n_d}$. Consider that a light slab of width $q$ is incident to more than one surface patches. Due to subsurface scattering, other surface patches that are not directly illuminated also produces non-zero reflectance. Suppose we sweep the light slab by translation so that it covers all visible patches on the surface. Let us assume for now that the effect of local diffuse $D$ and specular $I_s$ are absent, the summation of the $n_d$-th root of all captured images is:

$$qI_{v,1}(x)^{\frac{1}{n_d}} + qI_{v,2}(x)^{\frac{1}{n_d}} + ... + qI_{v,n}(x)^{\frac{1}{n_d}} = qI_{scat}(x)^{\frac{1}{n_d}} \tag{24}$$

which means that the image sum resulting by the sweeping light slab is simply a scaled-up version of the original $I_{scat}(x)$ of the images. Thus, as long as we can remove the contribution of $D$ and $I_s$, Eqn. (24) allows us to use a light slab to estimate $I_{scat}$, which is easier to produce in practice.

**Steps.** Our image acquisition procedure consists of the following steps.

1. First, we capture an image $I(x)$, which is illuminated by $\mathbf{L}$.
2. Then, we sweep a slab of rays parallel to $\mathbf{L}$ in constant speed, and capture a sequence of images $I_{seq,i}(x)$. $I(x)$ and $I_{seq,i}(x)$ are captured by the same digital video camera, using the same, fixed viewpoint.

After that, we set up the following equation:

$$CI(x)^{\frac{1}{n_d}} = D(x)^{\frac{1}{n_d}} + I_{seq,1}(x)^{\frac{1}{n_d}} + I_{seq,2}(x)^{\frac{1}{n_d}} + ... + I_{seq,p}(x)^{\frac{1}{n_d}} \quad (25)$$

where $p$ is total number of frames and $C = q \times g$. Variable $g$ is the velocity of the sweeping plane of rays. We have to consider this term because the speed of the plane may not be one patch per frame. Since Eqn. (25) only contains two unknowns, $n_d$ and $C$, we can choose $p \geq 2$ pixels $x$ from a region where specular effect is not significant to solve it, by using some standard optimization algorithm.

Then, we use simple thresholding to remove the contributions of $D$ and $I_s$ for each $I_{seq,i}$ since, by observation, the local diffuse and specular terms in the illuminated area are often much larger than the non-local subsurface scattering term in other area (Fig. 2(a)). The resulting hole after thresholding is shown in Fig. 2(b).

However, for $q > 1$, thresholding removes not only $D(x)$ and $I_s(x)$, but also some $I_{v,i}(x)$. Therefore, the missing $I_{v,i}(x)$ should be found. Since the response of the PSF for each $I_{seq,i}$ is small, quantization must produce some error. One suggestion to fill the hole is to treat each $I_{seq,i}$ as a 3D surface point with color values as $z$-coordinates. Then, the holes are filled by interpolation. Another suggestion is simple. If the width of light slab is small, the holes are just filled by the maximum color value surround the hole after thresholding. It still produces reasonable results. Alternatively, we can sweep another slab of rays in a direction *orthogonal* to the first sweeping slab. To recover the missing $I_{v,i}$, the following relation is used: $A \cup B = A + B - A \cap B$, where $A$ is the contribution to $I_{scat}$ from the first sweeping slab, and $B$ is the contribution to $I_{scat}$ from the second sweeping slab. The holes produced in the images from the first sweep can be filled in by the corresponding pixels in the images obtained after the second sweep. We call the resulting image an PSF image (Fig 2(c)).

Finally, we produce the $I_{scat}(x)^{1/n_d}$ by summing up the $n_d$-th root of the PSF images, and then scale the image sum by the $C$ we found earlier.

By using our proposed image acquisition method, not only $I_{scat,k}(x)$ can be robustly recovered for $k = 1, 2, 3$, but also $n_d$. These parameters are fed into the algorithm described in section 5, thus reducing the number of local and global parameters to four respectively: four global parameters $\{a_1, a_2, a_3, n_s\}$, and four local paramters $\{B(x), D_1(x), D_2(x), D_3(x)\}$. This makes our parameter estimation even more stable. Besides, the image acquisition step 1 and 2 mentioned above can be used to capture all the images $I_k(x)$.

## 7 Implementation and Results

We experiment our reflectance separation on real images, and use the separated components to perform novel view synthesis and highlight removal. Fig. 3 shows the image

(a) (b) (c)

**Fig. 2.** (a) The response of a mango pudding upon illumination by a light slab. This is $I_{seq,i}$. (b) Thresholded $I_{seq,i}$ to remove local diffuse and specular response. (c) The PSF image.

acquisition system we built. The video camera we used is Sony PC-100 Handycam. A thin light slab is produced by using a cardboard with a vertical slit set in the middle. Constant speed is maintained by moving the cardboard on a track, using a stepper motor with adjustable speed. During the capturing of $I_{seq,i}$, since the slab of rays is thin, the global radiance from the object is very small. Therefore, if we use normal exposure setting for regular image capture, the resulting PSF images will be very dark, and noise may dominate the image. Therefore, we use a different exposure setting for capturing $I_{seq,i}$.

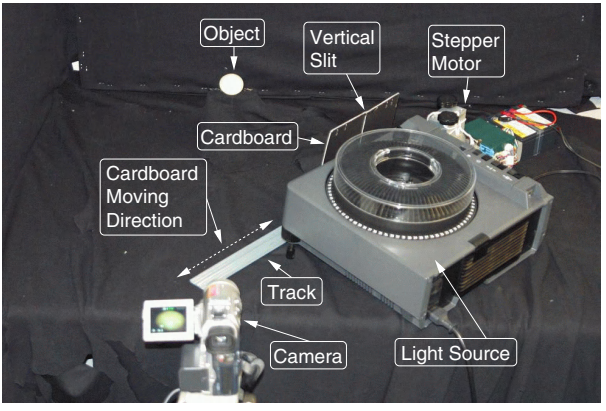



**Fig. 3.** Experimental setup.

When all the global and local parameters have been estimated using our method in section 5 and 6, we can reconstruct a novel image illuminated by any lighting vector, by setting the corresponding values of $\alpha_1, \alpha_2, \alpha_3$ and using Eqn. (21). We have conducted two experiments on our derived model with two different real objects: mango pudding and wax.

The mango pudding is located about 3 feet from the camera. The ray box can be placed very close to the object since it is set to approximate directional light source. The mango pudding is inside a thin and transparent plastic container which is hexagonal in

shape. The outer surface of the container is very rough so the effect of specular reflection is very small. There is a white paper label with the "ECCV04" logo on the front face. This label is cut up from a piece of white paper, which is translucent under illumination. The label has a different level of translucency compared with that of the pudding. The separated reflectance components for $k = 1$ are shown in Fig. 4(a-d)

The local diffuse component is very dark because the pudding is a highly scattering medium, almost all the energy has contributed to the non-directional subsurface scattering component. Since the paper label is a less scattering media, it is darker than the pudding in the separated $I_{scat}$. Although the container surface exhibits very little off-specular effect, this experiment shows that our method can still separate it out robustly.

Compared with the result produced by using the model of Lin and Lee [8], as shown in Fig. 4(e) and (f), where only BRDF is assumed, our result is much more reasonable. The off-specular component produced by their model covers the whole pudding because subsurface scattering is not modelled there.

Fig. 5 shows the same pudding re-lit by a novel lighting direction and intensity. Compared with the actual image, the synthesized image looks reasonable. Since the actual image is captured as regular images while our images $I_k$, $k = 1, 2, 3$, are captured by our new image acquisition method, they exhibit different intensity ranges. Therefore, some artifact can be noticed. It is suggested that all six images should be captured by our new image acquisition method, instead of the essential three images only, to improve the result.

Another object is a piece of wax. The surface of the wax is smooth, therefore the perfect mirror specular effect is very strong, which is not modelled in our reflectance representation. However, we want to test the robustness of our model. The separated reflectance components for $k = 1$ are shown in Fig. 6(a-d). The results produced using our model are similar to that of mango pudding. The local diffuse component $D$ is dark and the subsurface scattering component $I_{scat}$ is bright. Although our reflectance model does not model mirror specular component explicitly, the off-specular component approximates mirror specular effect very well.

In addition, Fig. 7 shows a novel image constructed by our method. Compared with the actual, ground truth image, the difference of the diffuse components (local and subsurface) is undistinguisable. The image difference is also shown. However, the specular components look quite different when compared with the specular components of the ground truth, which looks much brighter. It is due to mirror specular reflection. Besides, the edge of the wax shows some artifacts. This is because during image acquisition, we have to change the exposure setting of the camera by hand. The remote control of the camera does not provide this function. Some camera movement is unavoidable.

Although we use grey level images to separate reflectance, our separation method can be readily extended to color images. The color results are shown in the supplementary material.

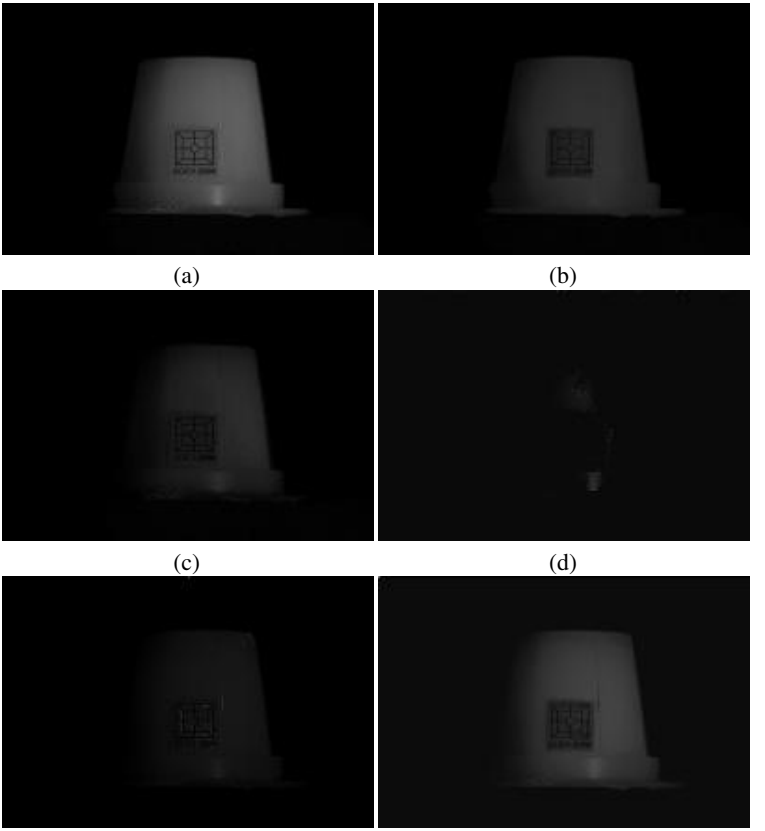



**Fig. 4.** Reflectance separation of a scattering medium mango pudding : (a) original image $I_k$, (b) non-directional subsurface scattering component $I_{scat,k}$, (c) local diffuse component $I_{local,k}$ , (d) off-specular component $I_{s,k}$, (e) diffuse component produced by using [8], (f) off-specular component produced by using [8]. We only show the case $k = 1$. See supplementary material of larger images, and for $k = 2, 3$.

**Fig. 5.** Novel image synthesis: (a) Real image, (b) synthetic image, (c) image difference between the real and the synthetic image. See supplementary material for larger images.

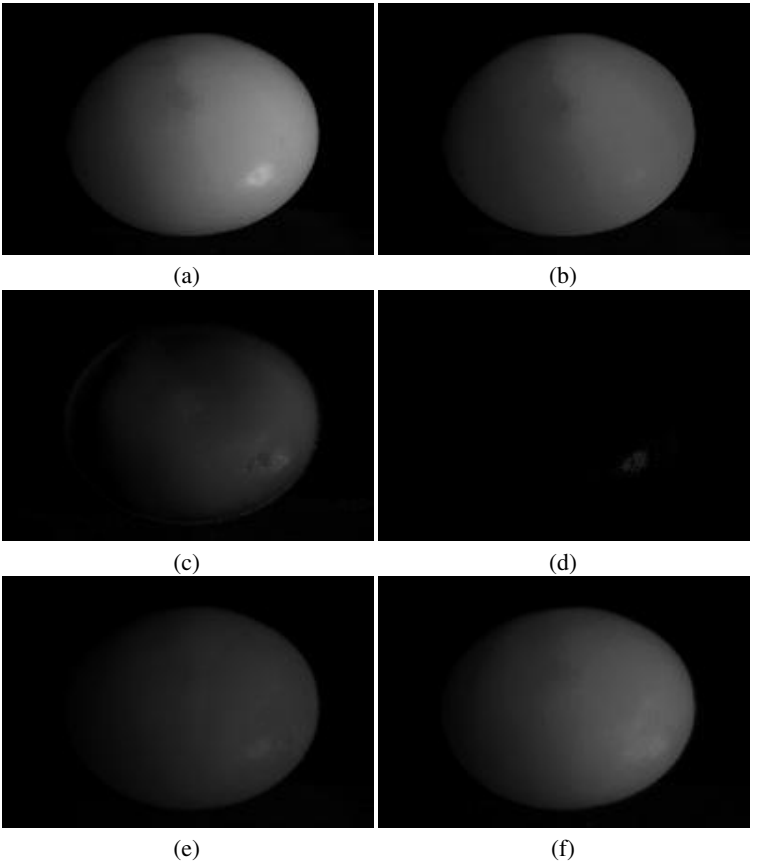



**Fig. 6.** Reflectance separation of a scattering medium wax : (a) original image $I_k$, (b) non-directional subsurface scattering component $I_{scat,k}$, (c) local diffuse component $I_{local,k}$ , (d) off-specular component $I_{s,k}$, (e) diffuse component produced by using [8], (f) off-specular component produced by using [8]. We only show the case $k = 1$. See supplementary material of larger images, and for $k = 2, 3$.

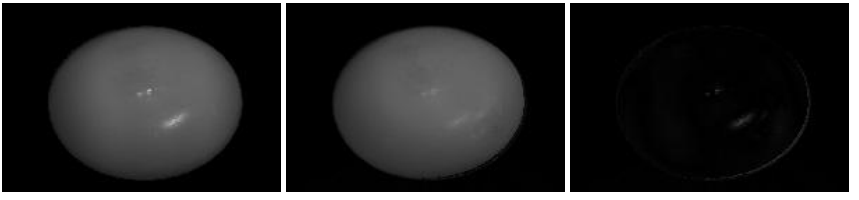



**Fig. 7.** Novel image: (a) Real image, (b) synthetic image, (c) image difference between the real and the synthetic image. See supplementary material for larger images.

## 8 Conclusion and Future Work

In this paper, we present an appearance-based model that allows for the separation of off-specular, non-Lambertian local diffuse and non-directional of subsurface scattering reflectance for real world objects. The BSSRDF model is necessary to capture subsurface scattering. A point spread function is estimated. Based on our mathematical derivation, a new and practical image acquisition method is proposed to capture non-directional subsurface scattering component, which complements and improves our parameter estimation process. Some successful reflectance separation experiments were conducted. Faithful novel images under different lighting condition can be generated by using our appearance model. We have also demonstrated improved result on hightlight removel in highly scattering medium, which is traditionally difficult for approaches based on BRDF. Since we do not model the effect of single scattering, our future work focuses on incorporating this component in order to support a wider range of translucent materials.

## References

1. R.L. Cook and K.E. Torrance. A reflectance model for computer graphics. *Computer Graphics*, 15:307–316, 1981.
2. P. Hanrahan and W. Krueger. Reflection from layered surfaces due to subsurface scattering. In *SIGGRAPH93*, pages 165–174, 1993.
3. A. Hertzmann and S.M. Seitz. Shape and materials by example: a photometric stereo approach. In *CVPR03*, pages I: 533–540, 2003.
4. H. W. Jensen, S. R. Marschner, M. Levoy, and P. Hanrahan. A practical model for subsurface light transport. In *SIGGRAPH01*, pages 511–518, 2001.
5. E. P. F. Lafortune, S. Foo, K. E. Torrance, and D. P. Greenberg. Non-linear approximation of reflectance function. In *SIGGRAPH97*, pages 117–126, 1997.
6. S. Lin and S. Lee. Estimation of diffuse and specular appearance. In *ICCV99*, pages 855–860, 1999.
7. S. Lin and S. Lee. A representation of specular appearance. In *ICCV99*, pages 849–854, 1999.
8. S. Lin and S.W. Lee. An appearance representation for multiple reflection components. In *CVPR00*, pages I: 105–110, 2000.
9. M. Oren and S.K. Nayar. Generalization of the lambertian model and implications for machine vision. *IJCV*, 14(3):227–251, April 1995.
10. Pierre Poulin and Alain Fournier. A model for anisotropic reflection. *Computer Graphics*, 24(4):273–282, August 1990.
11. H. Ragheb and E.R. Hancock. Highlight removal using shape-from-shading. In *ECCV02*, page II: 626 ff., 2002.
12. A. Shashua. Geometry and photometry in 3d visual recognition. In *MIT AI-TR*, 1992.

# Temporal Factorization vs. Spatial Factorization

Lihi Zelnik-Manor[1] and Michal Irani[2]

[1] California Institute of Technology, Pasadena CA, USA,
lihi@caltech.edu,
http://www.vision.caltech.edu/lihi
[2] Weizmann Institute of Science, Rehovot, Israel

**Abstract.** The traditional subspace-based approaches to segmentation (often referred to as multi-body factorization approaches) provide spatial clustering/segmentation by grouping together *points* moving with consistent *motions*. We are exploring a dual approach to factorization, i.e., obtaining temporal clustering/segmentation by grouping together *frames* capturing consistent *shapes*. Temporal cuts are thus detected at non-rigid changes in the shape of the scene/object. In addition it provides a clustering of the frames with consistent shape (but not necessarily same motion). For example, in a sequence showing a face which appears serious at some frames, and is smiling in other frames, all the "serious expression" frames will be grouped together and separated from all the "smile" frames which will be classified as a second group, even though the head may meanwhile undergo various random motions.

## 1 Introduction

The traditional subspace-based approaches to multi-body segmentation (e.g., [6, 7,9]) provide spatial clustering/segmentation by grouping *points* moving with consistent *motions*. This is done by grouping *columns* of the correspondence matrix of [17] (we review the definition in Section 1.1). In this work we show that to obtain *temporal* grouping of frames we cluster the *rows* of the same correspondence matrix instead of its columns. We show that this provides grouping of *frames* capturing consistent *shapes*, but not necessarily same motion. We further show that, to obtain such shape-based clustering of frames we need *not* develop any new segmentation/clustering scheme. We can use any of the existing algorithms suggested for clustering points (e.g., [6,7,9]). But, instead of applying them to the correspondence matrix as is, we apply them to its transpose.

Note, that spatial "multi-body factorization" [6,7,9] usually provides a highly sparse segmentation since commonly the number of points which can be tracked reliably along the sequence is low. Dense spatial segmentation requires dense optical flow estimation (e.g., [11]). In contrast, a small number of tracked points suffices to obtain a dense temporal clustering of frames, i.e., a classification of *all* the frames in the video clip. Furthermore, the dimensionality of the data, which is one of the major difficulties in spatial multi-body factorization, is significantly smaller for temporal segmentation. To obtain dense *spatial* factorization of the entire image (e.g., [11]), the number of points equals the number of pixels in the

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 434–445, 2004.
© Springer-Verlag Berlin Heidelberg 2004

image which can be extremely large (hundreds of thousands of pixels). This is *not* the case with *temporal* factorization. The number of frames in the video clip is usually only tens or hundreds of frames, and therefore the temporal factorization is *not* time consuming.

The standard approaches to temporal segmentation cut the video sequence into "scenes" or "shots", mainly by drastic changes in image appearance (e.g., [20,16,12]). Other approaches are behavior based (e.g., [19,15]) and segment the video into sub-sequences capturing different events or actions. The approach suggested here is fundamentally different and provides a temporal segmentation and clustering of frames which is based on non-rigid changes in shape. For example, in a sequence showing a face at some frames serious and in other frames smiling, all the "serious expression" frames will be grouped together and separated from all the "smile" frames which will be classified as a second group, even though the head may meanwhile undergo various random motions.

Our way of formulating the problem provides a unified framework for analyzing and comparing a number of previously developed independent methods. This new view of previous work is described in Section 4. For example, we show that the technique of Rui & Anandan [15] can be reformulated in terms of the factorization approach. Our analysis illustrates that their approach will detect cuts at large changes in motion, whereas we detect cuts at non-rigid shape changes. In a different work, Rao & Shah [14] suggested a view-invariant recognition method for complex hand movements. In Section 4 we show that the similarity constraint they use for *matching* shapes is equivalent to the one we use for *separating* between shapes.

We start by defining notations and reviewing the background to the multi-body factorization approach in Section 1.1. In Section 2 we present our approach to temporal factorization of shape and in Section 3 we explore its similarities to and differences from the standard spatial factorization of motion. As mentioned above, we review some related works in Section 4 and summarize in Section 5.

### 1.1 Background on Factorization Methods

Let $I_1, \ldots, I_F$ denote a sequence of $F$ frames with $N$ points tracked along the sequence. Let $(x_i^f, y_i^f)$ denote the coordinates of pixel $(x_i, y_i)$ in frame $I_f$ $(i = 1, \ldots, N\ ,\ f = 1, \ldots, F)$. Let $X$ and $Y$ denote two $F \times N$ matrices constructed from the image coordinates of all the points across all frames:

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_N^2 \\ & & \vdots & \\ x_1^F & x_2^F & \cdots & x_N^F \end{bmatrix} \qquad Y = \begin{bmatrix} y_1^1 & y_2^1 & \cdots & y_N^1 \\ y_1^2 & y_2^2 & \cdots & y_N^2 \\ & & \vdots & \\ y_1^F & y_2^F & \cdots & y_N^F \end{bmatrix} \tag{1}$$

Each row in these matrices corresponds to a single frame, and each column corresponds to a single point. Stacking the matrices $X$ and $Y$ of Eq. (1) vertically results in a $2F \times N$ "correspondence matrix" $W = \left[\frac{X}{Y}\right]$. It has been previously shown that under various camera and scene models [17,8,5] the correspondence matrix $W$ of a single object can be factorized into motion and shape matrices:

$W = MS$ (where $M$ and $S$ are low dimensional). When the scene contains multiple objects (see [6,7]) we still obtain a factorization into motion and shape matrices $W = MS$, where $M$ is a matrix containing the motions of all objects and $S$ is a block-diagonal matrix containing the shape information of all objects.

## 2 Temporal Factorization

The traditional subspace-based approaches to multi-body segmentation (e.g., [6,7,9]) provide *spatial* clustering of image points by grouping *columns* of the correspondence matrix $W = MS$. Note, that in the correspondence matrix $W$ every column corresponds to a point and every row corresponds to a frame. Thus, to obtain *temporal* clustering of frames we will apply clustering to the *rows* of $W$ instead of its columns. In this section we discuss the physical meaning of this temporal clustering of frames and suggest methods for obtaining it.

When factoring the correspondence matrix $W$ into motion and shape, the columns of the motion matrix $M$ span the columns of $W$ and the rows of the shape matrix $S$ span the rows of $W$. Hence, clustering the *columns* of $W$ into independent linear subspaces will group together points which share the same *motion*. Equivalently, clustering the *rows* of the correspondence matrix $W$ will group frames which share the same *shape*. Luckily, to obtain such row-based segmentation/clustering we need *not* develop any new segmentation/clustering scheme. We can use any of the existing algorithms suggested for segmenting/clustering columns (e.g., [6,7,9]). But, instead of applying them to $W$, we will apply them to $W^T$. We next show why this is true.

When the scene contains multiple ($K$) objects moving with independent motions, and the columns of $W$ are sorted to objects, then according to [6] the resulting shape matrix has a block diagonal structure:

$$W = [W_1, \ldots, W_K] = [M_1, \ldots, M_K] \begin{bmatrix} S_1 & & 0 \\ & \ddots & \\ 0 & & S_K \end{bmatrix} \tag{2}$$

where $W_i = M_i S_i$ is the correspondence matrix of the $i$-th object, with motion $M_i$ and shape $S_i$. The correct permutation and grouping of columns of $W$ into $W_1, \ldots, W_K$ to obtain the desired separation into independently moving objects was accordingly recovered [6,7] by seeking a block-diagonal structure for the shape matrix $S$. In other words, to obtain spatial segmentation of points we group the columns of $W$ into independent linear subspaces by assuming that $W$ can be factored into a product of two matrices, where the matrix on the right has a block diagonal form.

Now, taking the dual approach: When the sequence includes non-rigid shape changes ($Q$ independent shapes) and the *rows* of $W$ are sorted according to *shape*, then the resulting *motion* matrix has a block diagonal structure:

$$W = \begin{bmatrix} \tilde{W}_1 \\ \vdots \\ \tilde{W}_Q \end{bmatrix} = \begin{bmatrix} \tilde{M}_1 & & 0 \\ & \ddots & \\ 0 & & \tilde{M}_Q \end{bmatrix} \begin{bmatrix} \tilde{S}_1 \\ \vdots \\ \tilde{S}_Q \end{bmatrix} \tag{3}$$

The permutation and grouping of *rows* of $W$ into $\tilde{W}_1, \ldots, \tilde{W}_Q$ to obtain the desired separation into frames capturing independent *shapes* can therefore be obtained by seeking a block-diagonal structure for the *motion matrix* $M$.

Note, however, that if we now take the transpose of $W$ we get:

$$W^T = [\tilde{W}_1^T, \ldots, \tilde{W}_Q^T] = [\tilde{S}_1^T, \ldots, \tilde{S}_Q^T] \begin{bmatrix} \tilde{M}_1^T & & 0 \\ & \ddots & \\ 0 & & \tilde{M}_Q^T \end{bmatrix} \quad (4)$$

That is, the matrix $W^T$ can be factored into a product of two matrices where the matrix on the right is block diagonal. This is equivalent to the assumption made in the factorization of $W$ to obtain column clustering. Thus, we can use any of the algorithms suggested for segmenting/clustering columns (e.g., [6,7, 9]), however, instead of applying them to $W$ we will apply them to $W^T$. Our approach to subspace-based temporal clustering/factorization can therefore be summarized as follows:

Given a video clip of a dynamic scene:

1. Track reliable feature points along the entire sequence.
2. Place each trajectory into a column vector and construct the correspondence matrix $W = \left[\frac{X}{Y}\right]$ (see Eq. (1))
3. Apply any of the existing algorithms for column clustering (e.g., "multi-body factorization" of [6,7,9]), but to the matrix $W^T$ (instead of $W$).

Note, that when we say "independent shapes" we refer to independence between *rows* of different shape matrices (and *not* between columns/points). Independence between rows of two shape matrices occurs when at least part of the columns in those matrices are different. Recall, that the matrix $S$ corresponding to a rigid set of points is a $4 \times N$ matrix where each column holds the homogeneous coordinates $[X, Y, Z, 1]^T$ of a 3D point. Rigid shape changes can be viewed as the same set of points undergoing a different rigid motion, and therefore still have the same shape. However, *non-rigid* shape changes imply that some of the points move differently than others, i.e., some of the columns of the shape matrix change differently than others. This will lead to a different shape matrix and thus to assigning these frames to separate temporal clusters. Since every $4 \times N$ shape matrix has a row of 1's there is always partial linear dependence between shape matrices. To overcome that, we can use the Tomasi-Kanade [17] approach for removing the translational component by centering the centroid of the tracked points. Then the row of 1's is eliminated from the shape matrix, and we obtain full linear independence. Alternatively, some of the previously suggested approaches for sub-space segmentation can handle partial dependencies. In particular, we used the spectral clustering approach suggested in [13].

To illustrate this, Fig. 1 displays frames from a sequence showing a hand first open and then closed, while rotating and translating. As long as the hand is open, i.e., it's shape is not changing, the rows in the matrix $W$ will correspond to the same shape $\tilde{S}_{OPEN}$. However, the closing of the fingers im-

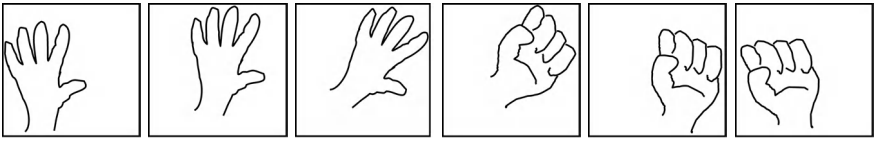


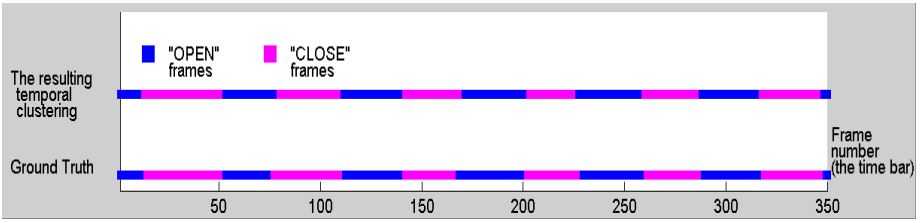

**Fig. 1.** An illustration of a hand rotating and translating while opening and closing its fingers. Frames (a),(b) and (c) capture the same shape (open hand) undergoing different rigid motion transformations. The closing of the fingers between frames (c) and (d) generates a new shape, independent of the previous one. The transformations between frames (d),(e),(f) can again be viewed as the same shape (closed hand) undergoing rigid motions.

plies a different shape of the object which cannot be represented as a rigid motion change. Instead we will obtain a new shape matrix $\tilde{S}_{CLOSE}$ so that: $W = \begin{bmatrix} \tilde{M}_{OPEN} & 0 \\ 0 & \tilde{M}_{CLOSE} \end{bmatrix} \begin{bmatrix} \tilde{S}_{OPEN} \\ \tilde{S}_{CLOSE} \end{bmatrix}$. Grouping the rows of $W$ is expected to group all the "OPEN" frames into one cluster and all the "CLOSE" frames into a separate cluster.

Fig. 2 shows this on a real video sequence. It further illustrates the difference between spatial segmentation/grouping of points based on *motion (column)* clustering, and temporal segmentation/grouping of frames based on *shape (row)* clustering. The sequence shows a hand opening and closing the fingers repeatedly. Feature points on the moving fingers were tracked along the sequence using the KLT tracker [10,1] and used to construct the correspondence matrix $W$. Factoring the rows of $W$ (i.e., the columns of $W^T$) into two clusters resulted in temporal shape-based segmentation of frames: It grouped together all the frames with fingers stretched open into one cluster, and all the frames with fingers folded into a second cluster (see Figs. 2.a,b,c). In contrast, applying the segmentation to the columns of $W$ resulted in spatial motion-based segmentation of points into independently moving objects: It grouped into one cluster the points on the fingers which moved mostly horizontally, and grouped into a second cluster points on the thumb which moved mostly vertically, (see Fig. 2.d). The palm of the hand was stationary and hence was ignored.

Fig. 3 displays another example of shape-based temporal segmentation. The video clip was taken from the movie "Lord of the Rings - Fellowship of the Ring", and shows two hobbits first relaxed and then screaming. Feature points were tracked along the sequence using the KLT tracker [10,1] and used to construct the correspondence matrix $W$. Grouping the rows of $W$ (columns of $W^T$) into two clusters detected the cut between the two expressions and grouped together all the "calm" frames separately from the "screaming" frames.

(a) Temporal factorization result:




(b) Example frame from the "OPEN" cluster

(c) Example frame from the "CLOSE" cluster

(d) Spatial factorization result

**Fig. 2.** Temporal vs. spatial clustering. (a) Results of temporal factorization (on the rows of $W$ = the columns of $W^T$) applied to a sequence showing a hand closing and opening the fingers repeatedly. Setting the number of clusters to 2 resulted in grouping all frames with fingers open into one cluster (marked in blue on the time bar) and all frames with fingers folded into a second cluster (marked in magenta on the time bar). Ground truth values, obtained manually, are shown for comparison. (b),(c) Example frames of the two temporal clusters. (d) Result of spatial factorization (on the columns of $W$) applied to the same sequence and the same tracked points. This grouped together all the points on the fingers (marked in red), which move mostly horizontally, and classified into a second cluster points on the thumb (marked in green) which move mostly vertically. Note, that since only sparse feature points were tracked along the sequence, the resulting spatial segmentation is highly sparse, whereas the resulting temporal factorization is dense (i.e., all the frames in the video sequence are classified) even though only a sparse set of points is used. Video can be found at http://www.vision.caltech.edu/lihi/Demos/TemporalFactorization.html

## 3 Comparing Temporal and Spatial Factorization

In this section we explore the major similarities and differences between the common motion based spatial factorization and our suggested approach to shape based temporal factorization.

**Data dimensionality:** One of the major difficulties in the multi-body factorization approach is the dimensionality of the data. As was shown by Weiss [18], the method of Costeira & Kanade [6] to multi-body segmentation is equivalent to applying spectral clustering to $W^T W$, which is an $N \times N$ matrix ($N$ being the number of points). If the number of points is large, then this is a very large matrix. Finding the eigenvectors of such a matrix (which is the heart of spectral clustering) is therefore extremely time consuming. To obtain dense *spa-*

*tial* factorization of the entire image (e.g., [11]), the number of points $N$ equals the number of pixels in the image which can be extremely large (hundreds of thousands of pixels).

However, this is *not* the case with *temporal* factorization. As explained in Section 2, to obtain temporal factorization of $W$, we apply the same algorithms suggested for spatial segmentation, but to $W^T$. In other words, this is equivalent to applying spectral clustering [18] to the matrix $WW^T$ (instead of $W^TW$). The dimension of $WW^T$ is $2F \times 2F$ , where $F$ is the number of frames in the video clip. Since $F << N$ ($F$ is usually only tens or hundreds of frames) , $WW^T$ is thus a small matrix, and therefore the temporal factorization is *not* time consuming. Furthermore, while *dense* spatial factorization requires dense flow estimation, dense temporal factorization can be obtained even if only a sparse set of reliable feature points are tracked over time. This is further explained next.

**Tracking sparse points vs. dense optical flow estimation:** Each column of $W$ contains the trajectory of a single point over time. The data in the matrix $W$ can be obtained either by tracking a sparse set of reliable points or by dense optical flow estimation. Since the spatial "multi-body factorization" clusters the *columns* of $W$, it will therefore classify only the points which have been tracked. Thus, when only a small set of reliable points is tracked, the resulting spatial segmentation of the image is sparse. Dense spatial segmentation of the image domain requires dense optical flow estimation. This, however, is not the case with temporal segmentation. Since our temporal factorization clusters the *rows* of $W$, there is no need to obtain data for all the points in the sequence. A sparse set of reliable points tracked through all the frames suffices for *dense* temporal factorization. This is because the number of columns in $W$ need not be large in order to obtain good row clustering. Results of temporal factorization using a small number of point tracks are shown in Figs. 2 and 3. In Fig. 4 we used dense optical flow measurements to show validity of the approach to both ways of obtaining data. Note, however, that even-though $N$ (the number of points) is large when using optical flow, the computational complexity of the temporal factorization is still low, since the size of $WW^T$ is independent of $N$ (it depends only on the number of frames $F$).

**Segmentation of $\left[\frac{X}{Y}\right]$ Vs. $[X,Y]$:** Let $W_v = \left[\frac{X}{Y}\right]$ and $W_h = [X,Y]$ where the subscript $v$ stands for vertical stacking of $X$ and $Y$ whereas the subscript $h$ stands for horizontal stacking of $X$ and $Y$. The common approaches to multi-body factorization (e.g., [6,7,9]) selected carefully tracked feature points, constructed the $W_v = \left[\frac{X}{Y}\right]$ matrix and clustered its columns. In this matrix each point has a single corresponding column, and each frame has two corresponding rows. Machline et al. [11] suggested applying multi-body factorization instead to the columns of $W_h = [X,Y]$. This allows to introduce directional uncertainty into the segmentation process, and thus enables dense factorization using unreliable points as well (i.e., dense flow). In this matrix, (i.e., $W_h$) each point has two corresponding columns whereas each frame has a single corresponding row. Thus, when clustering frames (rows) using temporal factorization it is simpler to use the matrix $W_h = [X,Y]$. Note, that when switching from $W_v = \left[\frac{X}{Y}\right]$ to $W_h = [X,Y]$ the motion matrix completely changes its structure whereas the shape matrix does not. Thus, in spatial multi-body factorization, which is

motion based, there is an inherent difference between the two approaches that leads to a different spatial segmentation when using $W_h = [X, Y]$ vs. $W_v = \left[\frac{X}{Y}\right]$ (see [11]). In contrast, the temporal factorization depends only on shape, thus applying temporal clustering either to $W_v = \left[\frac{X}{Y}\right]$ or to $W_h = [X, Y]$ will provide the same results. For simplicity we used the $W_h = [X, Y]$ matrix for temporal factorization and the $W_v = \left[\frac{X}{Y}\right]$ matrix for spatial clustering of points.

**Example:** Fig. 4 shows an example of shape vs. motion segmentation using dense optical flow estimation instead of sparse tracking data. The video clip was taken from the movie "Brave Heart", and shows the actor (Mel Gibson) first serious and then smiling while moving his head. The frame-to-frame optical flow was estimated using the robust optical flow estimation software of Michael Black [2] which is described in [4,3]. The frame-to-frame optical flow fields were composed over time to obtain flow-fields of all frames in the video clip relative to a single reference frame. These flow-fields were then organized in row vectors and stacked to provide the matrix $W_h = [X, Y]$. Applying spectral clustering to the rows of $W_h$ (i.e., applying factorization to the $F \times F$ matrix $W_h W_h^T$) separated the frames into two clusters: one cluster containing all the "smile" frames, and the other cluster containing all the "serious" frames (see Figs. 4.a,b). For comparison, applying the same clustering algorithm to the columns of $W_v$ (i.e., applying multi-body factorization to the $N \times N$ matrix $W_v^T W_v$) separated between regions with different motions (see Fig. 4.c).

**Summary:** For further clarification, we summarize in table 1 the observations made in Sections 2 and 3. This provides a summary of the comparison between spatial and temporal factorizations.

**Table 1.** Comparison summary of spatial factorization vs. temporal factorization

| | Spatial Factorization | Temporal Factorization |
|---|---|---|
| Apply clustering to | $W^T W$ | $W W^T$ |
| Data dimensionality | $N \times N$ | $F \times F$ |
| Data type | Points (columns) | Frames (rows) |
| Cluster by | Consistent motions | Consistent shapes |
| Sparse input | Sparse spatial segmentation | Dense temporal segmentation |
| Dense input | Dense spatial segmentation | Dense temporal segmentation |

## 4 A New View on Previous Work

In this section we show that our way of formulating the temporal factorization problem provides a unified framework for analyzing and comparing a number of previously developed independent methods.

The most related work to ours is that of Rui & Anandan [15] who used changes in the frame-to-frame optical flow field to segment activities into their fragments. Rui & Anandan [15] estimated the optical flow field between each pair of consecutive frames and stacked those into a matrix which is highly similar to our $W_h = [X, Y]$ matrix only with displacements instead of positions. They then applied SVD to the matrix, which provided the eigenflows spanning

(a) First temporal cluster ("calm" frames) (b) Second temporal cluster ("screaming" frames)

**Fig. 3.** Temporal clustering of frames. Results of temporal factorization (into 2 clusters) applied to a video clip taken from the movie "Lord of the Rings - Fellowship of the Ring". The clip shows two hobbits first calm and then screaming. The shape-based temporal factorization detected the cut between the two expressions and grouped together all the "calm" frames (some example frames are shown in column (a)) separately from all the "scream" frames (some example frames are shown in column (b)). Video can be found at http://www.vision.caltech.edu/lihi/Demos/TemporalFactorization.html

the space of all flow-fields and the coefficients multiplying these basis flow-fields. Temporal cuts were detected at *sign changes of those coefficients*. Their technique can be reformulated in terms of our temporal factorization approach. In our factorization into motion and shape one can view the shape matrix $S$ as being the eigen-vectors spanning the row space and $M$ being the coefficients multiplying these eigen-vectors. Looking at their work this way shows that they detect cuts at large changes in motion (e.g., shifting from clockwise rotation to counter-clockwise rotation), whereas we detect cuts at non-rigid shape changes and ignore the motion of each shape. Furthermore, reformulating [15] in terms of the temporal factorization approach allows extending it from simple *temporal segmentation* (i.e., detecting cuts) to *temporal clustering*.

Rao & Shah [14] suggested a view-invariant recognition method for complex hand movements. They first obtained hand trajectories (by tracking skin-colored regions) which were sorted according to general structure. Trajectories of similar structure were recognized as the same action by using a low-rank constraint on a matrix constructed from the tracks coordinates. This constraint is equivalent to the one we use for separating between shapes. We detect temporal cuts at

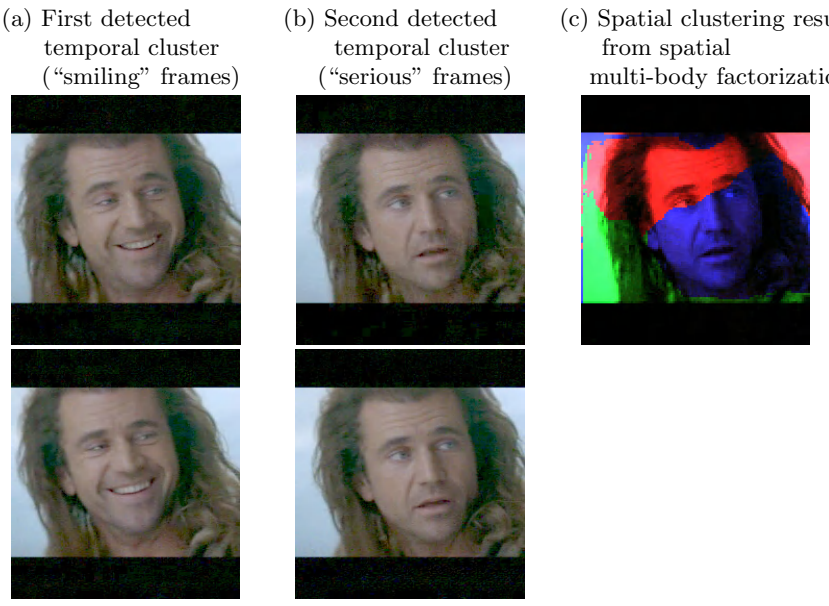



**Fig. 4.** Temporal vs. spatial clustering using dense optical flow. Results of factorization applied to a sequence taken from the movie "Brave Heart". The actor (Mel Gibson) is serious at first and then smiles while moving his head independently from his expression throughout the sequence. Optical flow was estimated relative to the first frame and the clustering was applied directly to it. We set the number of clusters to 2 for temporal factorization and to 3 for spatial factorization. (a) Sample frames from the first detected temporal cluster, all of which show the actor smiling. (b) Sample frames from the second detected temporal cluster which show the actor serious. (c) Since optical flow was used, we could obtain dense spatial segmentation. This separated between the forehead, the mouth region and a dangling group of hair. These correspond to three independent motions in the sequence: Along the sequence the actor raises his eyebrows and wrinkles his forehead. Independently of that the mouth region deforms when the actor smiles. The group of hair dingles as the head moves, again independently from the other two motions (the motion of the hair at the lower left part of the image can be seen in the frames in (a) and (b)). Video can be found at http://www.vision.caltech.edu/lihi/Demos/TemporalFactorization.html

increases of the rank and cluster the rows into groups of low rank, i.e., we group frames with the same (or similar) shape.

In a completely different context, Bregler et al. [5] obtained non-rigid object tracking using a factorization/subspace based approach. Their work is *not* related to neither spatial segmentation nor temporal factorization. Nevertheless, we found it appropriate to relate to their work since the shape matrix they used in their decomposition bares similarity to our shape matrix in Eq. (3), which can be misleading. There is a significant difference between their decomposition and ours. They assumed that the shape in each frame is a linear combination of all key-shapes whereas we associate a separate shape with each temporal cluster of frames.

## 5 Conclusions

We have explored the properties of temporal factorization of the correspondence matrix $W$ and its duality to spatial factorization of the same matrix. We showed that the temporal factorization provides a temporal segmentation and clustering of frames according to non-rigid changes in shape. This approach is unique in the sense that most existing temporal segmentation methods cut the video according to changes in appearance or changes in motion (as opposed to changes in shape).

We showed that to obtain temporal clustering we need *not* develop any new segmentation/clustering scheme but instead can utilize existing algorithms suggested for spatial segmentation. We further showed that dense spatial segmentation requires dense optical flow estimation whereas a small number of tracked points suffices to obtain a dense temporal clustering of frames, i.e., a classification of *all* the frames in the video clip. Furthermore, the dimensionality of the data, which is one of the major difficulties in spatial multi-body factorization, is significantly smaller for temporal segmentation.

The fact that the same factorization framework can be used for spatial segmentation and for temporal segmentation opens new possibilities that may lead to a combined approach for simultaneous spatio-temporal factorization.

**Acknowledgments.** This work was supported by the European Commission (VIBES project IST-2000-26001).

## References

1. S. Birchfield. Klt: An implementation of the kanade-lucas-tomasi feature tracker. http://robotics.stanford.edu/~birch/klt/.
2. M. J. Black. Dense optical flow: robust regularization. http://www.cs.brown.edu/people/black/.
3. M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *International Conference on Computer Vision*, pages 231–236, Berlin, Germany, 1993.
4. M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1):75–104, Jan. 1996.
5. C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shape from image streams. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 690–696, 2000.
6. J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In *International Conference on Computer Vision*, pages 1071–1076, Cambridge, MA, June 1995.
7. C.W. Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 2(29):133–150, 1998.
8. M. Irani. Multi-frame correspondence estimation using subspace constraints. *International Journal of Computer Vision*, 48(3):173–194, July 2002.
9. K. Kanatani. Motion segmentation by subspace separation and model selection. In *International Conference on Computer Vision*, volume 1, pages 301–306, Vancouver, Canada, 2001.

10. B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130, 1981.
11. M. Machline, L. Zelnik-Manor, and M. Irani. Multi-body segmentation: Revisiting motion consistency. In *Workshop on Vision and Modelling of Dynamic Scenes (With ECCV'02)*, Copenhagen, June 2002.
12. A. Nagasaka and Y. Tanaka. Automatic video indexing and full-video search for object appearances. In *Visual Databas Systems II, IFIP*, 1992.
13. A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *In Advances in Neural Information Processing Systems 14*, 2001.
14. C. Rao and M. Shah. Motion segmentation by subspace separation and model selection. In *International Conference on Computer Vision*, volume 1, pages 301–306, Vancouver, Canada, 2001.
15. Y. Rui and P. Anandan. Segmenting visual actions based on spatio-temporal motion patterns. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2000.
16. S. Swanberg, D.F. Shu, and R. Jain. Knowledge guided parsing in video databases. In *SPIE*, 1993.
17. C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9:137–154, November 1992.
18. Y. Weiss. Segmentation using eigenvectors: A unifying view. In *International Conference on Computer Vision*, pages 975–982, Corfu, Greece, September 1999.
19. L. Zelnik-Manor and M. Irani. Event-based video analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001.
20. H. Zhang, A. Kankanhali, and W. Smoliar. Automatic partitioning of full-motion video. In *Multimedia Systems*, 1993.

# Tracking Aspects of the Foreground against the Background

Hieu T. Nguyen and Arnold Smeulders

Intelligent Sensory Information Systems
University of Amsterdam, Faculty of Science
Kruislaan 403, NL-1098 SJ, Amsterdam, The Netherlands
tat,smeulders@science.uva.nl

**Abstract.** In object tracking, change of object aspect is a cause of failure due to significant changes of object appearances. The paper proposes an approach to this problem without a priori learning object views. The object identification relies on a discriminative model using both object and background appearances. The background is represented as a set of texture patterns. The tracking algorithm then maintains a set of discriminant functions each recognizing a pattern in the object region against the background patterns that are currently relevant. Object matching is then performed efficiently by maximization of the sum of the discriminant functions over all object patterns. As a result, the tracker searches for the region that matches the target object and it also avoids background patterns seen before. The results of the experiment show that the proposed tracker is robust to even severe aspect changes when unseen views of the object come into view.

## 1 Introduction

In visual object tracking, handling severe changes of viewpoint or object aspect has always been challenging. Change of aspect may be the result either of a self rotation of the tracked object or of a change of camera position. In either case, it is difficult to follow changing appearances of the object due to self-occlusion and disclosure at some parts of the object, and due to the lack of a reliable way for recovering the 3D motion parameters [1].

Current tracking methods handle viewpoint changes in two approaches: invariant-based and view-based. In the invariant-based approach, object matching is performed using appearance features invariant to viewpoint. The mean-shift tracking method [5], for example, uses histograms which are invariant to some degree of viewpoint change. Methods using a temporally smoothed and adaptive template also achieve some resistance to slight changes of object orientation [13,11]. The invariant-based methods, however, likely fail in case of severe changes of viewpoint, when a completely unseen side of the object moves into view.

View-based methods use considerably more a priori knowledge on the object. Many methods record a complete set of object views in advance [2,6,14]. An appearance model is then learned from this set to recognize any possible view of the object. The eigentracker by Black and Jepson [2], for example, extracts a few eigenimages from a set of object views. During tracking, the object region is localized simply by minimizing

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 446–456, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the distance to the subspace spanned by the eigenimages. The disadvantage of the view-based methods is that they need an a priori trained appearance model which is not always available in practice. Some other methods construct the view set online [12]. They store the key frames of the tracking results so as to recognize any previously seen object view when it appears again. There is no guarantee, however, that an unseen view can be identified. A fusion of offline and online learning of view information is proposed in [15].

This paper aims for robust tracking under severe changes of viewpoints in the absence of an a priori model. We achieve this using background information. This is based on the observation that even an unseen view of the object can still be identified if one can recognize the background and surrounding objects. It also conforms to a similar behavior of the human vision system where surrounding information is very important in localizing an object. Background has been used in tracking mainly via background subtraction, the well-known approach which works only for sequences with a stationary background. In case of a moving background, most current methods use the appearance information of the object only. Recent work by Collin and Liu [4] emphasizes the importance of the background appearance. The paper proposes to switch the mean-shift tracking algorithm between different linear combinations of the three color channels so as to select the features that distinguish the object most from the background. The features are ranked based on a variance test for the separability of the histograms of object and background. Improved performance compared to the standard mean-shift has been reported. Even so, color histograms have a limited identification power and the method appears to work only in the condition that the object appearance does not change drastically over the sequence. For high dimensional features like textures, the large number of combinations will be a problem for achieving real time performance.

In the presented approach, robustness to viewpoint change is attained by the discrimination of object textures from background textures. The algorithm should be working under a moving background.

Section 2 presents our discriminative approach for the target detection. The section discusses the representation of object appearance and how object matching is performed. Section 3 describes the tracking algorithm, the online training of object / background texture discriminant functions, and the updating of object and background texture templates. Section 4 shows the tracking results.

## 2 Discriminative Target Detection Using Texture Features

In the presented algorithm, the target object is detected by matching texture features. The locality and high discriminative power of theses features makes it easier to classify individual image patches as object or background.

### 2.1 Object Appearance Representation

Let us first consider the representation of object textures. Let $I(\boldsymbol{p})$ denote the intensity function of the current frame. Assume that the target region is mapped from a reference region $\Omega$ via a coordinate transformation $\varphi$ with parameters $\theta$. Object textures are then

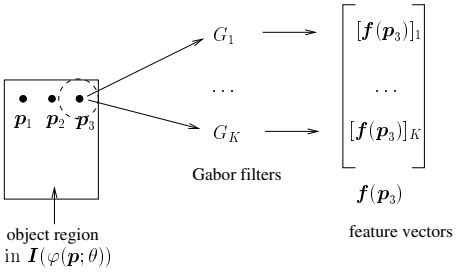



**Fig. 1.** *Illustration for the representation of object appearance.*

analyzed for the transformation compensated image $I(\varphi(\boldsymbol{p};\theta))$ using Gabor filters [10]. These filters have been used in various applications for visual recognition [7,9] and tracking [3]. Each pair of Gabor filters has the form:

$$G_{symm}(\boldsymbol{p}) = \cos\left(\frac{\boldsymbol{p}}{r}\cdot\boldsymbol{n}_\nu\right)\exp\left(-\frac{\|\boldsymbol{p}\|^2}{2\sigma^2}\right)$$
$$G_{asymm}(\boldsymbol{p}) = \sin\left(\frac{\boldsymbol{p}}{r}\cdot\boldsymbol{n}_\nu\right)\exp\left(-\frac{\|\boldsymbol{p}\|^2}{2\sigma^2}\right) \tag{1}$$

where $\sigma, r$ and $\nu$ denote the scale, the central frequency and orientation respectively, and $\boldsymbol{n}_\nu = \{\cos(\nu), \sin(\nu)\}$. Setting these parameters to different values creates a bank of filters. Denote them $G_1, \ldots, G_K$. Object texture at pixel $\boldsymbol{p} \in \Omega$ is characterized by vector $\boldsymbol{f}(\boldsymbol{p}) \in \mathbb{R}^K$ which is composed of the response of image $I(\varphi(\boldsymbol{q};\theta))$ to the Gabor filters:

$$[\boldsymbol{f}(\boldsymbol{p})]_k = \sum_{\boldsymbol{q}\in\mathbb{R}^2} G_k(\boldsymbol{p}-\boldsymbol{q})I(\varphi(\boldsymbol{q};\theta)) \tag{2}$$

where $[\boldsymbol{f}(\boldsymbol{p})]_k$ denotes the $k^{th}$ component of $\boldsymbol{f}(\boldsymbol{p})$, $1 \le k \le K$. When necessary, we also use the notation $f(\boldsymbol{p};\theta)$ to explicitly indicate the dependence of $\boldsymbol{f}$ on $\theta$. The appearance of a candidate target region is represented by the ordered collection of the texture vectors at $n$ sampled pixels $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n \in \Omega$, see Figure 1:

$$\mathcal{F} = \{\boldsymbol{f}(\boldsymbol{p}_1), \ldots, \boldsymbol{f}(\boldsymbol{p}_n)\} \tag{3}$$

As $\boldsymbol{f}(\boldsymbol{p})$ governs the information of an entire neighborhood of $\boldsymbol{p}$, there is no need to compute the texture vector for all pixels in $\Omega$. Instead, $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n$ are sampled with a spacing.

### 2.2 Object Matching

The target detection amounts to finding the parameters $\theta$ that give the optimal $\mathcal{F}$. This is based on the two criteria:

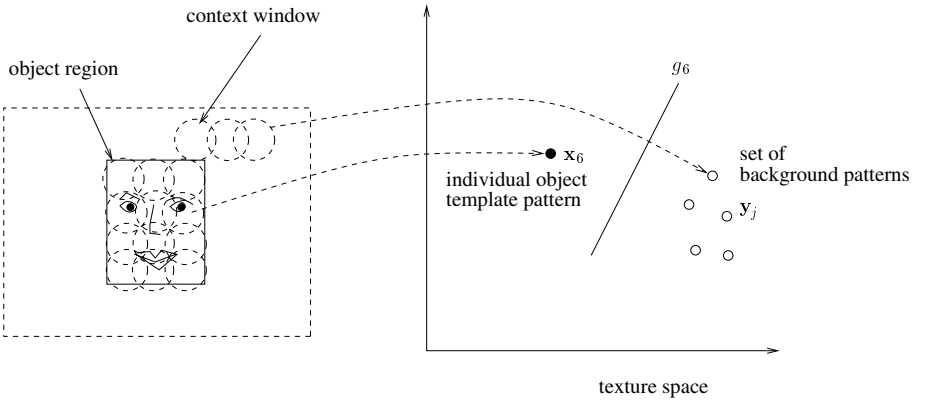



**Fig. 2.** *Illustration of the target detection using object/background texture discrimination.*

1. The similarity between $\mathcal{F}$ and a set of object template features:

$$\mathcal{O} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\},\ \mathbf{x}_i \in \mathbb{R}^K \tag{4}$$

   There is a correspondence between the vectors in $\mathcal{F}$ and $\mathcal{O}$, that is, $\boldsymbol{f}(\boldsymbol{p}_i)$ should match $\mathbf{x}_i$, since both of them represent the texture at pixel $\boldsymbol{p}_i$. This valuable information is ignored in the related approach [4], as it is based on histogram matching. The object templates are updated during tracking to reflect the most recent object appearance.
2. The contrast between $\mathcal{F}$ and a set of background template features:

$$\mathcal{B} = \{\mathbf{y}_1, \ldots, \mathbf{y}_M\},\ \mathbf{y}_j \in \mathbb{R}^K \tag{5}$$

   These are the texture vectors of background patterns observed so far in a context window surrounding the object, see Figure 2. The modelling of the background through a set of local patterns is mainly to deal with the difficulty in the construction of a background image. It is desired that every $\boldsymbol{f}(\boldsymbol{p}_i)$ is distinguished from all $\mathbf{y}_j$. As the background moves, $\mathcal{B}$ is constantly expanded to include new appearing patterns. On the other hand, a time-decaying weighting coefficient $\alpha_j$ is associated with every pattern $\mathbf{y}_j$. The coefficient enables the tracker to forget patterns that have left the context window.

We optimize $\mathcal{F}$ by maximizing the sum of a set of local similarity measures each computed for one vector in $\mathcal{F}$:

$$\max_{\theta} \sum_{i=1}^{n} g_i(\boldsymbol{f}(\boldsymbol{p}_i; \theta)) \tag{6}$$

Here, $g_i(\boldsymbol{f}(\boldsymbol{p}_i; \theta))$ is the local similarity measure for object texture at pixel $\boldsymbol{p}_i$. We choose $g_i$ to be a linear function:

$$g_i(\boldsymbol{f}) = \boldsymbol{a}_i^T \boldsymbol{f} + b_i \tag{7}$$

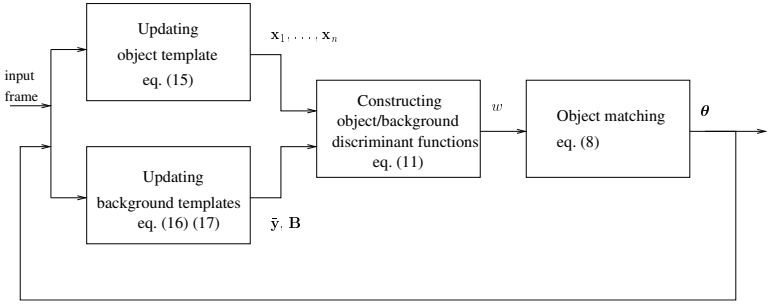



**Fig. 3.** *The illustration of the tracking algorithm.*

where $\boldsymbol{a}_i \in \mathbb{R}^K, b_i \in \mathbb{R}$ are the parameters. Furthermore, to satisfy the two mentioned criteria, $g_i$ is chosen to be a discriminant function. Specifically, $g_i$ trained to respond positively when $\boldsymbol{f} = \mathbf{x}_i$ and negatively when $\boldsymbol{f} \in \mathcal{B}$, see Figure 2. Note that in case where $\boldsymbol{f}(\boldsymbol{p}_i; \theta)$ represents an unseen object pattern, it may not match $\mathbf{x}_i$ but does not belong to $\mathcal{B}$ either. In this case, $g_i(\boldsymbol{f}(\boldsymbol{p}_i; \theta))$ likely has a value around zero, which is still higher than a mismatch. As such, by avoiding the background patterns the tracker is still able to find the correct object even in case of an aspect change.

In eq. (6), only the directions $\boldsymbol{a}_i$ matter. The value of $b_i$ does not affect the maximization result. Using eq. (2), eq. (6) is rewritten as:

$$\max_{\theta} \sum_{\boldsymbol{q}} I(\varphi(\boldsymbol{q}; \theta)) w(\boldsymbol{q}) \tag{8}$$

where

$$w(\boldsymbol{q}) = \sum_{i=1}^{n} \sum_{k=1}^{K} a_{ik} G_k(\boldsymbol{p}_i - \boldsymbol{q}) \tag{9}$$

and $a_{ik}$ denotes the $k^{th}$ component of $\boldsymbol{a}_i$. As observed, (8) is the inner product of image $I(\varphi(\boldsymbol{q}; \theta))$ and function $w$. In particular, if only the translational motion is considered, $\varphi(\boldsymbol{q}; \theta) = \boldsymbol{q} + \theta$, and hence, object matching boils down to the maximization of the convolution of the current frame $I(\boldsymbol{q})$ with function $w$ which is regarded as the target detection kernel.

## 3 Algorithm Description

Based on the matching method described above, we propose a tracking algorithm whose data flow diagram is given in Figure 3. This section addresses the issues that remain, including the construction of discriminant functions $g_i$ and the updating of object and background templates.

### 3.1 Construction of Object / Background Discriminant Functions

In principle, any linear classifier from pattern recognition can be used for training $g_i$. However, in view of the continuous growth of the set of background patterns, the selected classifier should allow for the training in the incremental mode, and should be computationally tractable in real time tracking. To this end, we adapt the LDA (Linear Discriminant Analysis) [8]. Function $g_i$ minimizes the cost function:

$$\min_{\boldsymbol{a}_i, b_i} (\boldsymbol{a}_i^T \mathbf{x}_i + b_i - 1)^2 + \sum_{j=1}^{M} \alpha_j (\boldsymbol{a}_i^T \mathbf{y}_j + b_i + 1)^2 + \frac{\lambda}{2} \|\boldsymbol{a}_i\|^2 \tag{10}$$

over $\boldsymbol{a}_i$ and $b_i$. The weighting coefficients $\alpha_j$ are normalized so that $\sum_{j=1}^{M} \alpha_j = 1$. The regularization term $\frac{\lambda}{2}\|\boldsymbol{a}_i\|^2$ is added in order to overcome the numerical instability due to high-dimensionality of texture features. The solution of eq. (10) is obtained in closed form:

$$\boldsymbol{a}_i = \kappa_i [\lambda \mathbf{I} + \mathbf{B}]^{-1} [\mathbf{x}_i - \bar{\mathbf{y}}] \tag{11}$$

where

$$\bar{\mathbf{y}} = \sum_{j=1}^{m} \alpha_j \mathbf{y}_j \tag{12}$$

$$\mathbf{B} = \sum_{j=1}^{m} \alpha_j [\mathbf{y}_j - \bar{\mathbf{y}}][\mathbf{y}_j - \bar{\mathbf{y}}]^T \tag{13}$$

$$\kappa_i = \frac{1}{1 + \frac{1}{2}[\mathbf{x}_i - \bar{\mathbf{y}}]^T [\lambda \mathbf{I} + \mathbf{B}]^{-1} [\mathbf{x}_i - \bar{\mathbf{y}}]} \tag{14}$$

As observed, the discriminant functions depend only on the object templates $\mathbf{x}_i$, the mean vector of background textures $\bar{\mathbf{y}}$ and the covariance matrix $\mathbf{B}$. These quantities can efficiently be updated during tracking.

Note that the background is usually non-uniform and therefore its textures are hardly represented just by one mean pattern $\bar{\mathbf{y}}$. Instead, the diversity of background patterns is encoded in the covariance matrix $\mathbf{B}$.

### 3.2 Updating Object and Background Templates

As we are dealing with sequences with severe viewpoint changes, the object templates need to be updated constantly to follow up varying appearances. On the other hand, a hasty updating is sensitive to sudden tracking failure and stimulates template drift. So, the updated template should be a compromise between the latest template and the new data. For this purpose, sophisticated temporal smoothing filters have been proposed [13]. In this work, however, for the implementation simplicity, we use the simple averaging filter:

$$\mathbf{x}_i^{(t)} = (1-\gamma)\mathbf{x}_i^{(t-1)} + \gamma \boldsymbol{f}(\boldsymbol{p}_i;\theta) \tag{15}$$

where the superscript $(t)$ denotes the time, and $0 < \gamma < 1$ is a predefined coefficient.

During object motion, constantly new patterns enter the context window and some other patterns leave the window. The background representation should be updated accordingly. It would be difficult to track reliably a background pattern from the moment of entering until its leaving. So, we keep all the observed patterns and gradually decrease the coefficients $\alpha_j$ that control the influence of the patterns in eq. (10). In this way, the tracker can forget the outdated patterns.

At every tracking step, the Gabor filters are applied for image $I(\boldsymbol{p})$ at $m$ fixed locations in the context window, yielding $m$ new background texture vectors denoted $\mathbf{y}_{M+1}, \cdots, \mathbf{y}_{M+m}$. The weighting coefficients are then distributed over the new and old elements in $\mathcal{B}$ so that the total weight of the new patterns amounts to $\gamma$ while that of the old patterns is $1-\gamma$. Therefore, each new pattern is assigned an equal weighting coefficient $\alpha_j = \gamma/m$. Meantime, the coefficient of every existing pattern in $\mathcal{B}$ is re-scaled with the factor $1-\gamma$. Let $\bar{\boldsymbol{y}}_{new} = \frac{1}{m}\sum_{j=M+1}^{M+m} \mathbf{y}_j$. The update equations for $\bar{\mathbf{y}}$ and $\boldsymbol{B}$ are:

$$\bar{\mathbf{y}}^{(t)} = (1-\gamma)\bar{\mathbf{y}}^{(t-1)} + \gamma \bar{\boldsymbol{y}}_{new} \tag{16}$$

$$\boldsymbol{B}^{(t)} = (1-\gamma)\boldsymbol{B}^{(t-1)} + (1-\gamma)\bar{\mathbf{y}}^{(t-1)}\bar{\mathbf{y}}^{(t-1)\,T} - \bar{\mathbf{y}}^{(t)}\bar{\mathbf{y}}^{(t)\,T} + \frac{\gamma}{m}\sum_{j=M+1}^{M+m} \mathbf{y}_j\mathbf{y}_j^T \tag{17}$$

## 4 Experiment

We have performed several experiments to verify the ability of the proposed tracking algorithm in handling severe viewpoint changes. In the current implementation, only translational motion is considered. For the extraction of texture features, the algorithm uses a set of twelve Gabor filters created for scale $\sigma = 4$ and $r = 2.0$ and six directions of $\nu$ equally spaced by $30^o$. The target region is set to a rectangle. Object pixels $\boldsymbol{p}_1, \cdots, \boldsymbol{p}_n$ are sampled with a spacing of 4 pixels between each other in both horizontal and vertical axes. The same spacing is applied for the background pixels in the context window. For the updating of the object and background texture templates, we have set the weighting coefficient $\gamma = 0.2$.

For comparison, we also applied an intensity SSD tracker using an adaptive template. In every frame this algorithm recalculates the template as a weighted average between the latest template and new intensity data, where the weight of the new data is $\gamma = 0.2$. This averaging results in a smoothed template which is also resilient to viewpoint changes in some degree. Unlike the proposed approach, this algorithm does not use background information.

Figure 4 shows an example of head tracking. Initially the head is at the frontal view pose. The background is non-uniform, and the camera is panning back and forth, keeping the head in the center. The guy turns to the sides and even to the back, showing completely

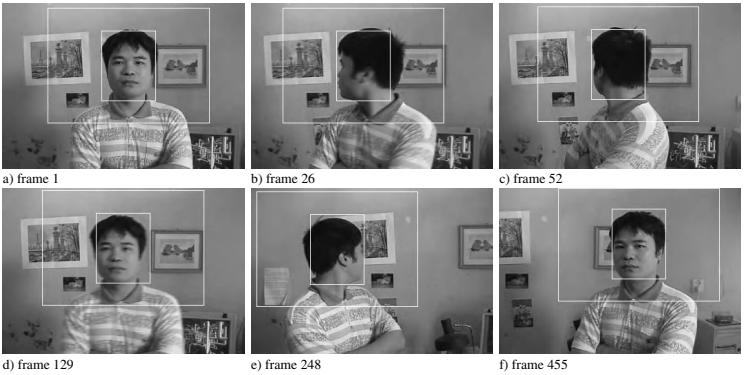



**Fig. 4.** *Head tracking results under severe viewpoint changes by the proposed algorithm. The outer rectangle indicates the context window.*

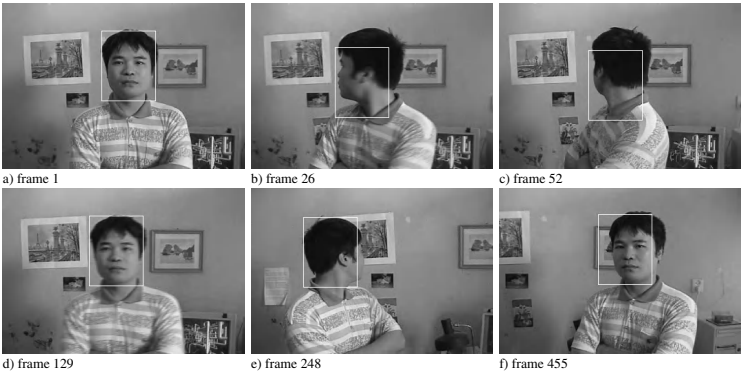



**Fig. 5.** *Tracking results for the same sequence in Figure 4 by the SSD tracker using an adaptive template.*

different views of the head. As observed, the proposed tracker could capture even the back view of the head which is unseen previously and is rather different from the initial frontal view. Figure 5 shows the tracking results for the same sequence but with the SSD tracker. This tracker also exhibited a robust performance under slight pose changes of the head, but it gave wrong results when the head pose changed severely as in Figure 5b and c. Nevertheless, the SSD tracker did not lose track and well recovered from the drift when the head returned back to the frontal view. This success can be explained by the uniqueness of the black hair in the scene.

A clear example where the proposed algorithm outperforms the SSD tracker is shown in Figure 6 and Figure 7. The figures show the tracking results by the two trackers respectively for a sequence where a mousepad is rotated around its vertical axis, switching between the blue front side and the completely black back side. As we expected, the SSD tracker drifted off at the first transition of view, see Figure 7b. This is easily explained by the similarity between the color of the front side of the mousepad and the color of the

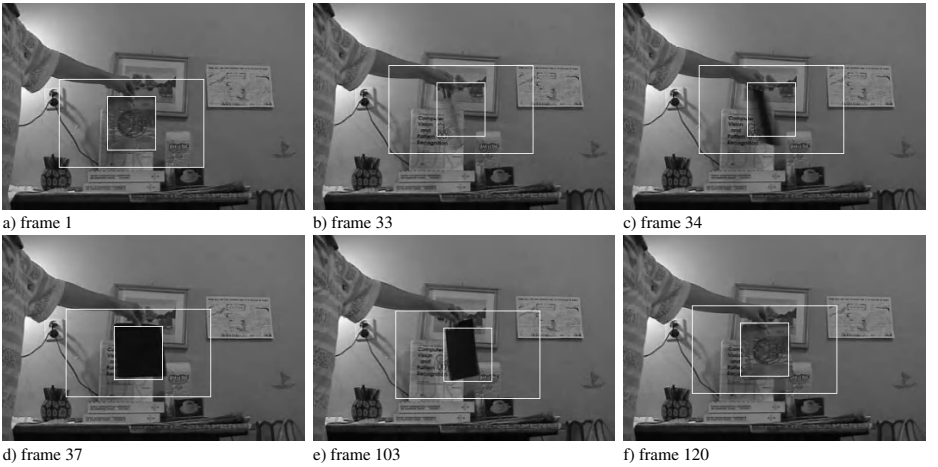



**Fig. 6.** *Tracking results by the proposed algorithm.*

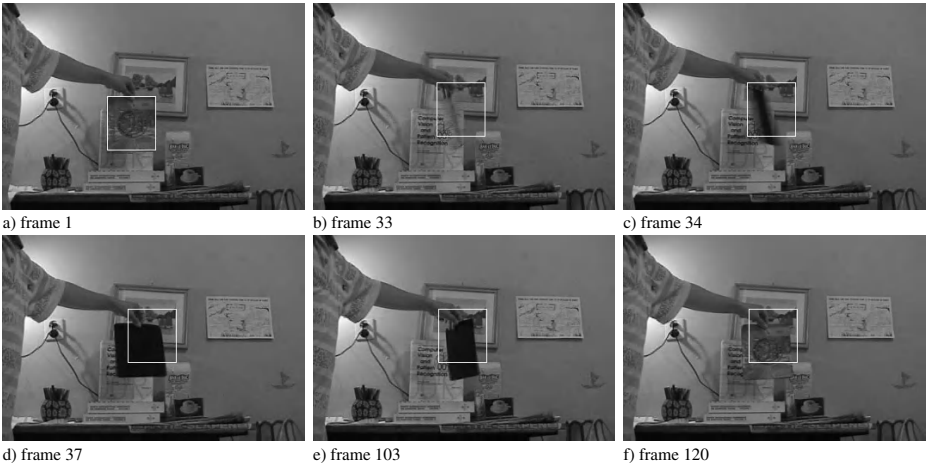



**Fig. 7.** *Tracking results for the sequence in Figure 6 by the SSD tracker using an adaptive template.*

wall. In contrast, the proposed algorithm recovered perfectly when the unseen dark side comes into view, see Figure 6d. It could also successfully lock back on the front side as in Figure 6f. The results prove that the proposed tracker rather chooses an unseen object region instead of a background region.

Figure 8 shows another head tracking result by the proposed algorithm for a movie clip. In this sequence, the camera pans fast to the left. The background is cluttered and contains several other moving objects. The results show the success of the proposed algorithm in tracking the head through several severe pose changes, as well as the robustness to the background motion and clutter.

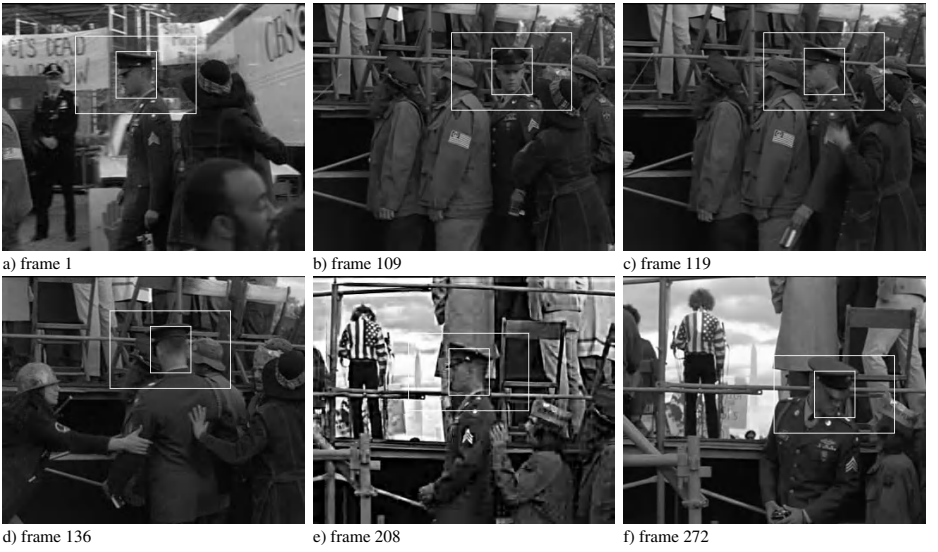



**Fig. 8.** *Tracking results by the proposed algorithm with a fast moving and cluttered background.*

## 5 Conclusion

The paper has shown the advantage of the background information for object tracking under severe viewpoint changes, especially when an unseen aspect of the object emerges. We have proposed a new tracking approach based on the discrimination of object textures from background textures. The high dimensionality of texture features allows for a good separation between the two scene layers. While the representation of the background by a set of patterns is robust to background motion, weighting the patterns in a time-decaying manner allows to get rid of outdated patterns. The algorithm keeps track of a set of discriminant functions each separating a pattern in the object region from the background patterns. The target is detected by the maximization of the sum of the discriminant functions, taking into account the spatial distribution of object texture. The discriminative approach prevents the tracker from accepting background patterns, and therefore enables the tracker to identify the correct object region even in case of substantial changes in object appearance.

For future work, we plan to improve several issues. We plan to test other more sophisticated classifiers to improve the accuracy of the target detection. The algorithm can also be extended to the multiscale mode with the propagation of the tracking result through the scales of the Gabor filters. Finally, more accurate models for the representation and updating of object and background template patterns will be considered.

## References

1. G. Adiv. Inherent ambiguities in recovering 3-D motion and structures from noisy flow field. *IEEE Trans. on PAMI*, 11(5):477–489, 1989.

2. M. J. Black and A. D. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. In *Proc. of European Conf. on Computer Vision*, pages 329–342, 1996.
3. O. Chomat and J.L. Crowley. Probabilistic recognition of activity using local appearance. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages II: 104–109, 1999.
4. R. Collins and Y. Liu. On-line selection of discriminative tracking features. In *Proc. IEEE Conf. on Computer Vision*, 2003.
5. D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In *CVPR00*, pages II:142–149, 2000.
6. T.F. Cootes, G.V. Wheeler, K.N. Walker, and C.J. Taylor. View-based active appearance models. *Image and Vision Computing*, 20(9-10):657–664, 2002.
7. J.G. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. on PAMI*, 15(11):1148–1161, 1993.
8. P. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley, New York, 2001.
9. S. Gong, S. J. McKenna, and Collins J. J. An investigation into face pose distributions. In *Proc. of 2nd Inter. Conf. on Automated Face and Gesture Recognition*, Killington, Vermont, 1996.
10. A.K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.
11. A.D. Jepson, D.J. Fleet, and T.F. El-Maraghi. Robust online appearance models for visual tracking. In *Proc. IEEE Conf. on Computer Vision and Pattern Recogn., CVPR01*, 2001.
12. L.P. Morency, A. Rahimi, and T. Darrell. Adaptive view-based appearance models. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages I: 803–810, 2003.
13. H.T. Nguyen, M. Worring, and R. van den Boomgaard. Occlusion robust adaptive template tracking. In *Proc. IEEE Conf. on Computer Vision, ICCV'2001*, pages I: 678–683, 2001.
14. S. Ravela, B.A. Draper, J. Lim, and R. Weiss. Tracking object motion across aspect changes for augmented reality. In *ARPA Image Understanding Workshop*, pages 1345–1352, 1996.
15. L. Vacchetti, V. Lepetit, and P. Fua. Fusing online and offline information for stable 3D tracking in real-time. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages II: 241–248, 2003.

# Example-Based Stereo with General BRDFs

Adrien Treuille[1], Aaron Hertzmann[2], and Steven M. Seitz[1]

[1] University of Washington, Seattle, WA, USA
{treuille, seitz}@cs.washington.edu
[2] University of Toronto, Toronto, ON, Canada
hertzman@dgp.toronto.edu

**Abstract.** This paper presents an algorithm for voxel-based reconstruction of objects with general reflectance properties from multiple calibrated views. It is assumed that one or more reference objects with known geometry are imaged under the same lighting and camera conditions as the object being reconstructed. The unknown object is reconstructed using a radiance basis inferred from the reference objects. Each view may have arbitrary, unknown distant lighting. If the lighting is calibrated, our model also takes into account shadows that the object casts upon itself. To our knowledge, this is the first stereo method to handle general, unknown, spatially-varying BRDFs under possibly varying, distant lighting, and shadows. We demonstrate our algorithm by recovering geometry and surface normals for objects with both uniform and spatially-varying BRDFs. The normals reveal fine-scale surface detail, allowing much richer renderings than the voxel geometry alone.

## 1 Introduction

Recovering an object's geometry from multiple camera orientations has long been a topic of interest in computer vision. The challenge is to reconstruct high-quality models for a broad class of objects under general conditions. Most prior multiview stereo algorithms assume Lambertian reflectance—the radiance at any point is the same in all directions—although progress has begun on relaxing this assumption, e.g. [1,2,3,4]. Also, virtually all approaches to multiview stereo ignore non-local light phenomena, such as interreflections and cast shadows. Because these simplifying assumptions do not hold for large classes of objects, new algorithms are needed.

In this paper, we present an approach for the volumetric reconstruction of objects with *general reflectance*. That is, we do not assume a particular BRDF model. Instead, we assume that one or more reference objects of related material are observed under the same conditions as the object being reconstructed. Our experimental setup handles surfaces with isotropic BRDFs, although a more general setup could in theory be used to capture anisotropic BRDFs as well. Our algorithm also handles varying camera and light positions: we assume only that the cameras and lights are distant and separated from the object by a plane. In addition, we show how to account for cast shadows in voxel coloring. We note

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 457–469, 2004.
© Springer-Verlag Berlin Heidelberg 2004

that cast shadows occur when a voxel is not "visible" to a light source. In this way, we can extend voxel coloring's treatment of visibility to handle shadows as well.

Our approach is based on the *orientation-consistency* cue which states that, under orthographic projection and distant lighting, two surface points with the same surface normal and material exhibit the same radiance. This cue was introduced in the context of photometric stereo [5]. We adapt orientation-consistency to multiview reconstruction within the voxel coloring framework [6]. We chose voxel coloring mainly for simplicity, although orientation-consistency could be used with other stereo algorithms as well. Besides handling general BRDFs, orientation-consistency enables computing per-voxel normals, which is not possible in conventional voxel coloring. These normals reveal fine surface details that the voxel geometry alone does not capture.

## 2 Related Work

Most previous work in multiview stereo focuses on diffuse objects, e.g. [7,8,9,10, 11], although progress has recently been made on treating other types of surfaces. In particular, recent work has addressed the case of completely specular objects. Zheng and Murata [12] reconstruct purely specular objects; Bonfort and Sturm [13] and Savarese and Perona [14] study the case where the surface is a mirror.

Stereo methods have also been proposed for diffuse-plus-specular surfaces. Carceroni and Kutulakos [1] reconstruct moving objects assuming a Phong model with known specular coefficient. They propose an ambitious, complex procedure combining discrete search and nonlinear optimization. Jin *et al.* [2] exploit a BRDF constraint implied by a diffuse-plus-specular model which requires a fixed light source. Yang *et al.* [3] adapt the space carving algorithm for a diffuse-plus-specular model by making heuristic assumptions on the BRDF and illumination; they assume that all observations lie on a line in color space. They also require a fixed light source and color images. All of these algorithms make some diffuse-plus-specular assumption on the reflectance. Our technique improves on these in that we consider a broader class of BRDFs. On the other hand, these techniques do not require a reference object.

The only known stereo method that handles completely general BRDFs is that of Zickler *et al.* [4], which exploits Helmholtz reciprocity, and demonstrates high quality results. Their method applies to perspective projection, and does not need a reference object. However, they require a constrained illumination setup with point light sources and cameras reciprocally placed, whereas we allow arbitrary, distant cameras and illumination. Finally, they do not take into account shadows, though we believe our shadow technique can be used for Helmholtz stereo as well.

Our algorithm also handles completely general BRDFs, by using a cue called *orientation-consistency*. This cue, proposed by Hertzmann and Seitz for photometric stereo, has been shown to work for BRDFs as complex as velvet and brushed fur [5]. Other techniques in photometric stereo also handle non-Lambertian

BRDFs, e.g. [15,16,17]. However, all of these photometric stereo approaches are constrained by solving only for a normal map, which can lead to geometric distortions and makes depth discontinuities troublesome. Instead of just a normal map, we solve for a full object model with normals.

Our work integrates orientation-consistency into the the voxel coloring algorithm [6], which was originally designed for Lambertian surfaces. The algorithm handles visibility, but requires constraints on camera placement. Space carving [7] relaxes the constraints on camera placement. We adapt voxel coloring to apply to general BRDFs, and show how voxel coloring's treatment of visibility can be extended to handle shadows. We also adapt the camera placement constraints to orthographic projection. Generalizing our work to space carving is straightforward; we chose voxel coloring because it is simpler than space carving, and provides a testbed for evaluating the novel features presented in this paper.

While voxel coloring uses voxels to represent the geometry, level sets are becoming an increasingly popular geometric representation in computer vision. In particular, Faugeras and Keriven [11] variationally solve for a level set describing the geometry. We believe our technique could be applied to level-set stereo by integrating orientation-consistency into the objective function; other diffuse multiview stereo algorithms could also benefit from orientation-consistency.

Most previous multiview stereo methods do not explicitly handle cast shadows. One exception is the work of Savarese *et al.* [18] who demonstrate a volumetric carving technique that uses only shadows. Their work differs from ours in that they assume shadows can be detected *a priori*, and in that they do not make use of reflectance information.

## 3 Reconstructing Objects with a Single Material

We now show how we adapt voxel coloring to reconstruct objects with general BRDFs. The central component of voxel coloring is the *photo-consistency* test, which determines if a voxel is consistent with the input photographs. Conventional voxel coloring uses a test that is suitable only for Lambertian surfaces. We replace this test with *orientation-consistency*, which applies to general BRDFs.

We begin with a brief summary of voxel coloring. The target object must be photographed to satisfy the *ordinal visibility constraint*. This condition ensures that there exists a traversal order of the voxels so that occluding voxels are always visited before those that they occlude. For example, in our experiments the camera is placed above the target object, and we traverse the voxels layer-by-layer from top to bottom. A consistency test is applied to each voxel $\nu$ in turn. If $\nu$ is deemed *consistent* with all input images, then it is included in the volumetric model. Otherwise, $\nu$ is *inconsistent* and is discarded. The consistency test only considers views in which the voxel is not occluded (see Section 3.3).

### 3.1 Diffuse Photo-consistency

We call the consistency metric of the original voxel coloring algorithm *diffuse photo-consistency*. The test consists of projecting the voxel $\nu$ into each input image to produce a vector $\mathbf{V}_\nu$ of intensities

$$\mathbf{V}_\nu = [I_{1,\nu}, I_{2,\nu}, \ldots, I_{n,\nu}]^T \tag{1}$$

which we call a target *observation vector*. For color images, we have separate observation vectors $\mathbf{R}_\nu$, $\mathbf{G}_\nu$, and $\mathbf{B}_\nu$ for the red, green, and blue channels, respectively. These can be concatenated into the *color observation vector*:

$$\mathbf{V}_\nu = \left[\mathbf{R}_\nu^T, \mathbf{G}_\nu^T, \mathbf{B}_\nu^T\right]^T. \tag{2}$$

The voxel is consistent if all intensities are nearly the same, measured by testing whether the sum of the variance in intensity over all color channels falls below a specified threshold. The algorithm ensures that only views in which the voxel is visible are considered during the variance test.

### 3.2 Orientation-Consistency

Diffuse photo-consistency is physically meaningful for Lambertian surfaces, but does not apply to more general BRDFs, since we cannot assume that the radiance from a voxel will be the same in all directions. Instead, we propose an example-based consistency metric called *orientation-consistency*, which is adapted from [5].

To begin, assume that the target object consists of only one material and that a reference object of the same material and with known geometry has been observed in all the same viewpoints and illuminations as the target object. Moreover, assume the reference object exhibits the full set of visible normals for each camera position. Now consider a voxel $\nu$ on the surface of the target object. There must exist a point $\mathbf{p}$ on the reference object that has the same normal, and, thus, the same observation vector. Hence, we can test consistency by checking if such a point $\mathbf{p}$ exists.

In practice, we densely sample the surface of the reference object and project these sampled points $\mathbf{p}_1, \ldots, \mathbf{p}_k$ into the input images to form set of reference observation vectors $\mathbf{V}_{\mathbf{p}_1}, \ldots, \mathbf{V}_{\mathbf{p}_k}$. We use this database of reference observation vectors to determine if a given target observation vector is consistent with some normal. Formally, a voxel $\nu$ with observation vector $\mathbf{V}_\nu$ is *orientation-consistent* if there exists a point $\mathbf{p}_i$ such that

$$\frac{1}{d}||\mathbf{V}_\nu - \mathbf{V}_{\mathbf{p}_i}||^2 < \epsilon \tag{3}$$

for some user-determined threshold $\epsilon$. Division by $d$, the number of dimensions in the vectors, gives the average squared error. We note that $d$ may differ for different observation vectors: when a target voxel $\nu$ is occluded or in shadow for

some camera, the corresponding dimensions of the target and reference observation vectors are excluded.

We note some special features of orientation-consistency. First, unlike diffuse photo-consistency, orientation-consistency is not limited to diffuse BRDFs. In fact, the technique handles arbitrary isotropic BRDFs. Moreover, by finding the exact reference point $\mathbf{p}_i$ that minimizes Equation (3), we can assign the corresponding surface normal to each consistent voxel. Later, we can render these normals, as in bump mapping [19], to reveal much more visual detail than the voxel geometry itself contains.

### 3.3 Orthographic Ordinal Visibility

Our algorithm depends crucially on being able to determine for which cameras a voxel is visible. Seitz and Dyer showed that visibility could be determined if the voxels are traversed so that occluding voxels are always processed before those they occlude. This implies an *ordinal visibility constraint* on camera placement for perspective projection [6]. We use the same approach, but adapted for orthographic projection.

Suppose we have $n$ cameras pointing in directions $\mathbf{c}_1, \ldots, \mathbf{c}_n$, where each of the $\mathbf{c}_i$ are unit vectors. Our constraint on camera placement is that there exists a vector $\mathbf{v}$ such that $\mathbf{v} \cdot \mathbf{c}_i > 0$ for each camera direction $\mathbf{c}_i$ (Fig. 1). Informally, we can think of a plane with normal $\mathbf{v}$ separating the cameras from the object, and we iterate over voxels by marching in the direction of $\mathbf{v}$. More formally, the order is defined so that we visit point $\mathbf{a}$ before point $\mathbf{b}$ if $(\mathbf{b} - \mathbf{a}) \cdot \mathbf{v} > 0$.

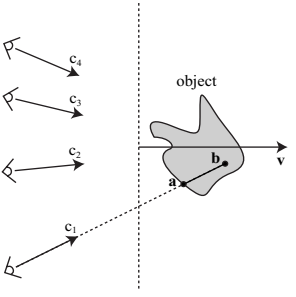



**Fig. 1.** Orthographic Ordinal Visibility: A plane with normal $\mathbf{v}$ separates the camera from the scene. Each point $\mathbf{a}$ is processed before any point $\mathbf{b}$ that it occludes.

To see why this inequality processes the points in the correct order, suppose that some point $\mathbf{a}$ occludes a point $\mathbf{b}$ for camera direction $\mathbf{c}_i$. Then $\mathbf{b} - \mathbf{a} = \alpha \mathbf{c}_i$ with $\alpha > 0$. Applying a dot product with $\mathbf{v}$ we get $(\mathbf{b} - \mathbf{a}) \cdot \mathbf{v} = \alpha \mathbf{c}_i \cdot \mathbf{v}$. Since $\mathbf{c}_i \cdot \mathbf{v}$ is positive by assumption, $(\mathbf{b} - \mathbf{a}) \cdot \mathbf{v} > 0$ and $\mathbf{a}$ is processed before $\mathbf{b}$, by definition of the ordering.

### 3.4 Single Material Results

We now describe our experimental setup and present results. Our target object was a soda bottle, and our reference object was a snooker ball. Both objects were spray-painted with a shiny green paint to make the materials match. We then photographed the objects under three different lighting conditions from a fixed camera, while varying the object orientation with a turntable. The photographs were taken with a zoom lens at 135mm, and the light sources were placed at least five meters away. This setup approximates an orthographic camera and directional lighting, which are the assumptions of orientation-consistency. The camera was calibrated using a freely-available toolbox [20].

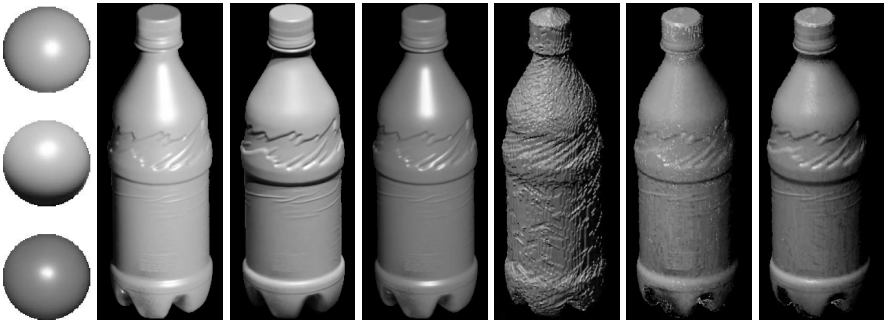



**Fig. 2.** (a) Reference spheres shown for the 3 different illuminations. (b) Corresponding input images for one of 30 object orientations. (c) Voxel reconstruction using three lights. (d) Rendered normals obtained from one light source. (e) Rendered normals obtained from three light sources.

Our input consisted of 30 object orientations with 3 illumination conditions each, for a total of 90 input images. Using fewer than 3 lights adversely affects the recovered normals, though it does not seem to affect the geometry. We also note that, by symmetry, the appearance of the sphere does not change as it rotates on the turntable. Therefore we need only one image of the sphere per illumination condition.

Fig. 2(a) shows input images of the reference sphere for the 3 different lighting conditions. Fig. 2(b) shows corresponding input images of the bottle. The voxel reconstruction can be seen in Fig. 2(c). Finally, Fig. 2(d) and 2(e) show the model with the recovered normals. The improvement of 2(e) over 2(d) is achieved by using multiple illuminations.

The ability to recover normals is one of the strengths of our algorithm as it reveals fine-grained surface texture. The horizontal creases in the label of the bottle are a particularly striking example of this: the creases are too small to be captured by the voxels, but show up in the normals. Fig. 3 shows different views rendered from the reconstruction.

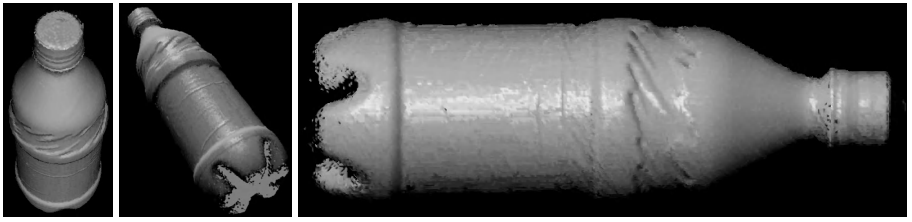


**Fig. 3.** Views of the bottle reconstruction.

## 4 Generalizations

While the technique described in the previous section makes voxel coloring possible for general BRDFs, it suffers from several limitations. First, our consistency function does not take into account shadows that the object casts upon itself. In addition, the model works only in the restrictive case that the target object has a single BRDF shared by the reference object. In this section, we show how these assumptions are relaxed.

### 4.1 Handling Shadows

Unlike previous work in multiview stereo, our framework allows the illumination to vary arbitrarily from image to image. Given this setup, it is almost inevitable that, in at least some views, a complex object will cast shadows on itself. We address this phenomenon by treating shadowed voxels as if they were occluded, that is, by excluding the shadowed dimensions from the observation vector. Thus, we need a method to determine if a voxel is in shadow.

We note that shadows occur when a voxel is not "visible" to a light source, and we can therefore compute shadows exactly as we do occlusions. We assume that the light directions are calibrated. Just as each image has an *occlusion mask* aligned with the camera (see [6]), we now say that each image also has a *shadow mask* aligned with the light. Voxels can be projected onto this mask as if the light were a camera. The occlusion and shadow masks are initially empty. When a voxel is deemed consistent, it is projected onto all occlusion masks and all shadow masks. The set of pixels to which the voxel projects, it's *footprint* is marked to exclude these regions from future computation; subsequent voxels projecting onto marked regions are considered occluded or in shadow, as the case may be. Of course, this technique implies that voxels casting shadows must be processed before those which they shadow, but this is equivalent to the corresponding visibility requirement, and the same theory applies; both the cameras *and* lights must be placed so that their directions satisfy the ordinal visibility constraint of Section 3.3.

Note that this technique of detecting shadows is completely independent of the choice of consistency test, and can be implemented with other forms of voxel coloring. In particular, by using perspective projection and the conventional ordinal visibility constraint [6], point light sources can be handled.

### 4.2 Varying BRDFs

Our second generalization relaxes the assumption that the target and reference objects consist of the same single material. Instead, we use a *basis* of reference objects with BRDFs related to that of the target object. As in [5], we assume that the colors observed on the target object can be expressed as a linear combination of observation vectors from the reference objects.

As in Section 3.4, we use spheres as reference objects. Suppose that $\mathbf{p}_1, \ldots, \mathbf{p}_k$ is a set of points *in sphere surface coordinates*, so that we may talk about the same point on multiple spheres. For every point $\mathbf{p}_i$ there are $s$ observation vectors $\mathbf{V}^1_{\mathbf{p}_i}, \ldots, \mathbf{V}^s_{\mathbf{p}_i}$, one for each of the $s$ spheres, which we concatenate into an *observation matrix*:

$$\mathbf{W}_{\mathbf{p}_i} = \left[\mathbf{V}^1_{\mathbf{p}_i}, \ldots, \mathbf{V}^s_{\mathbf{p}_i}\right]. \tag{4}$$

We assume a voxel is consistent if it can be explained by some normal and some material in the span of the reference spheres. Formally, a voxel $\nu$ with observation vector $\mathbf{V}_\nu$ is orientation-consistent with respect to the reference spheres if there exists a point $\mathbf{p}_i$ and a *material index* $\mathbf{m}$ such that

$$\frac{1}{d}||\mathbf{V}_\nu - \mathbf{W}_{\mathbf{p}_i}\mathbf{m}||^2 < \epsilon \tag{5}$$

for a user-specified threshold $\epsilon$. As in Equation (3), $d$ is the number of dimensions left after deleting all occluded and shadowed images from the observation vectors. Note that while we find the best point $\mathbf{p}_i$ by linearly searching through our database of samples, the best material index $\mathbf{m}$ can be directly computed for each $\mathbf{p}_i$ using the pseudo-inverse $(^+)$ operation:

$$\mathbf{m} = (\mathbf{W}_{\mathbf{p}_i})^+ \mathbf{V}_\nu. \tag{6}$$

In summary, the algorithm is as follows: for each target voxel, we test consistency by iterating over all possible source points $\mathbf{p}_i$. For each source point, we compute the optimal material index $\mathbf{m}$. If any pair $(\mathbf{p}_i, \mathbf{m})$ satisfies Equation (5), then the voxel $\nu$ is added to the reconstruction; otherwise it is discarded.

### 4.3 Multiple Material Results

We now present experimental results for the generalized algorithm. For the experiments, we photographed two spheres, one matte gray, and the other specular black. We decomposed the gray sphere into a red, green, and blue diffuse basis; the black sphere was used to handle specularities. To calibrate the lights for the shadow computation, we photographed a mirrored ball and estimated the reflection of the light direction by computing the centroid of the brightest pixels.

We reconstructed two target objects. The first was a porcelain cat. Fig. 4(a) shows an input photograph. Fig. 4(b) is the voxel reconstruction, and Fig. 4(c) shows the reconstruction rendered with the recovered normals. Other views of the reconstruction are shown in Fig. 4(d). The black diagonal line across the

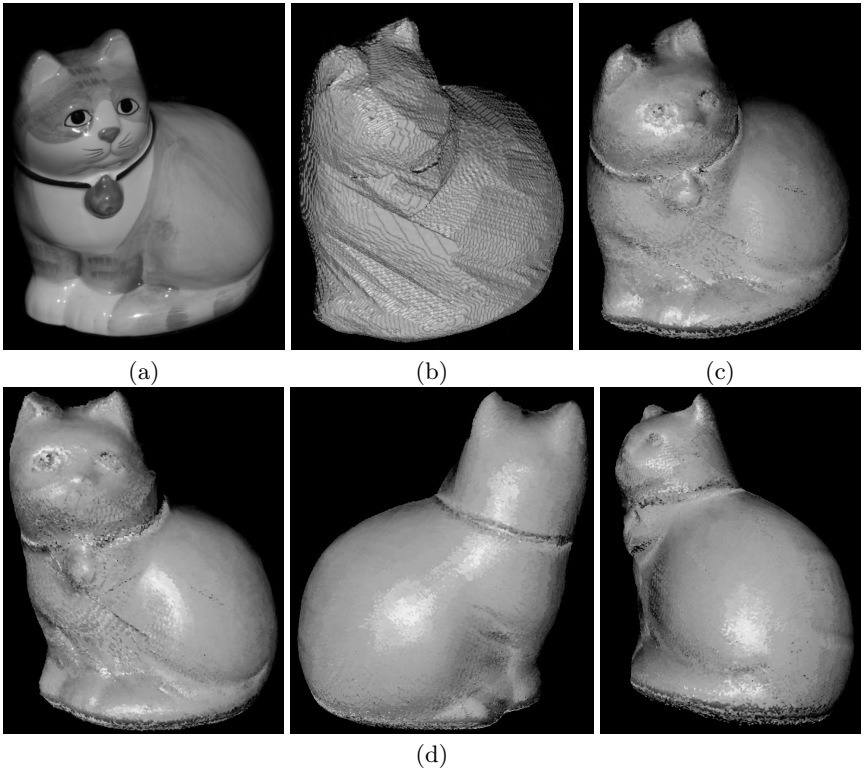



**Fig. 4.** Cat model. (a) Input image. (b) Voxel reconstruction. (c) Rendered with normals. (d) New views.

front of the cat in Fig. 4(d) is an artifact caused by insufficient carving. The true surface lies several voxels behind the recovered surface, but the algorithm has not been able to carve that far. Setting the consistency threshold lower would have prevented this artifact, but caused overcarving in other parts of the geometry. In both this model and the next, overestimation of the geometry added some noise to the normals, because the algorithm was trying to fit normals to points not on the surface. We discuss this issue further in Section 5.

The second target object was a polished marble rhinoceros. As with the cat, Fig. 5(a) shows an input photograph; Fig. 5(b) is the voxel reconstruction, and Fig. 5(c) shows the reconstruction with the recovered normals. Note that the normals reveal the three chisel marks on the side of the rhino in Fig. 5(c). Finally, Fig. 5(d) shows additional views.

We now highlight two aspects of the technique. First, to show that our algorithm can correctly carve past the visual hull, Fig. 6 shows a close-up of the hind legs of the rhino model. Looking, in particular, at the gap between the legs, we can see that the geometry is better approximated by our algorithm (Fig. 6(c)) than by the visual hull alone (Fig. 6(a)), even for this relatively untextured re-

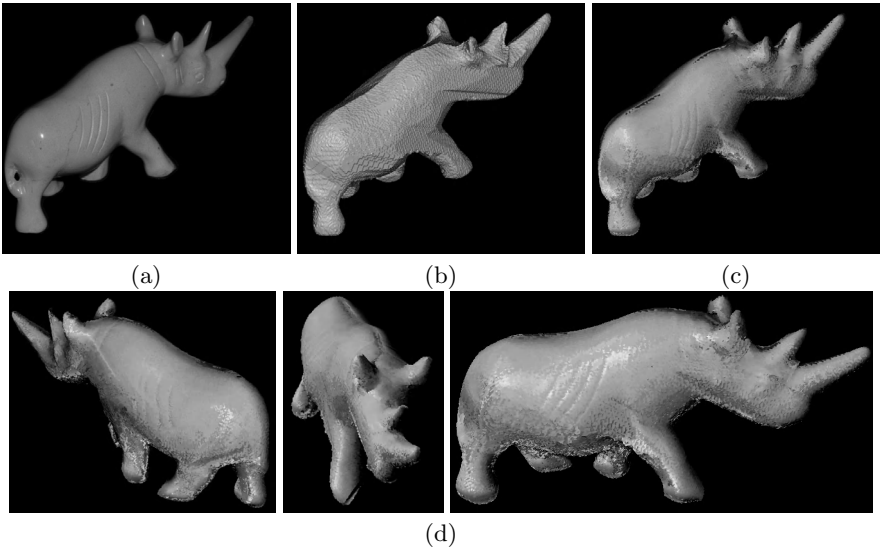



**Fig. 5.** Rhino Model. (a) Input image (one of 120). (b) Voxel reconstruction. (c) Rendered with normals. (d) New views.

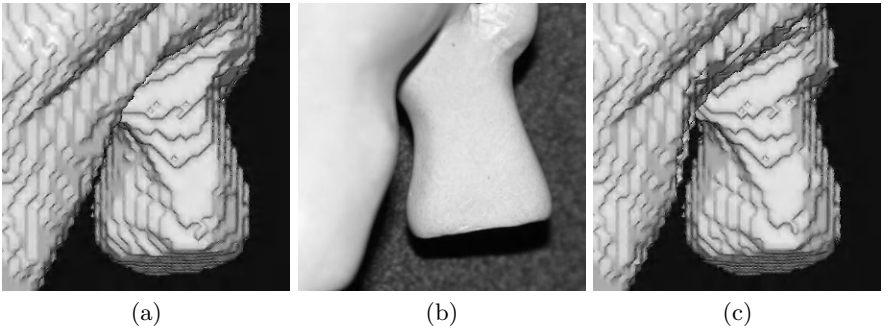
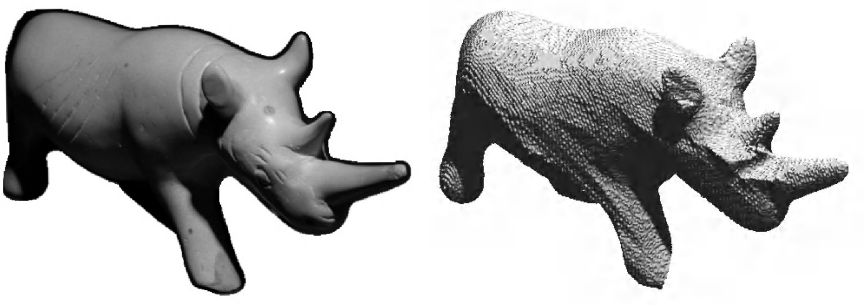



**Fig. 6.** Detail of the hind legs of the rhino model. Note the gap between the legs. (a) Visual Hull. (b) Photograph not from input sequence. (c) Reconstructed model.

gion. Note that the comparison photograph, Fig. 6(b), could not have been used as input, as it would have violated the ordinal visibility constraint. To carve this area, we used a lower consistency threshold than that used in Fig. 5, which resulted in overcarving of other parts of the model. We expect that an adaptive threshold technique like [9] could address this problem.

The second aspect we highlight is the shadowing technique. Specifically, Fig. 7 provides a visualization of the shadows. Fig. 7(a) shows one input image from the rhino sequence, and Fig. 7(b) shows the shadow voxels computed for that image. Note the shadows cast by the right ear, and those on the right fore and hind legs.

These show that we are getting a good approximation of the shadows. However, the shadows would improve further if the geometry were better estimated.



(a) (b)

**Fig. 7.** (a) Input image. (b) Shadows (dark regions) computed for this image.

## 5 Discussion and Future Work

This paper presented a novel volumetric reconstruction algorithm based on orientation-consistency. We assume only that the cameras and lights are distant, and separated from the object by a plane. We showed how to integrate orientation-consistency into voxel coloring; other multiview techniques would have been possible as well. Our ability to solve for normals within the voxel framework yields a dramatic improvement in the representation of fine-scale surface detail. Although our experimental setup is designed for isotropic BRDFs, a more general setup could be used for anisotropic BRDFs, as was shown in [5] for the photometric case.

We also showed how voxel coloring can be adapted to handle cast shadows if the lights are calibrated. As a step toward this end, we adapted the ordinal visibility constraint to the orthographic case. We view our treatment of shadows as a first step toward integrating non-local lighting phenomena into traditional reconstruction techniques.

The main difficulty we encountered is carving past the visual-hull. Two issues may be simultaneously contributing to this problem. First, voxel coloring is hampered by trying to find a *consistent* model, which can be weaker than minimizing an error metric on the model. Second, our algorithm works better for a single material than for multiple materials, possibly because the target materials are not sufficiently well modeled by the reference materials. As a result, the consistency threshold must be set high, and too few voxels are carved. Better results may be possible using orientation-consistency in conjunction with more recent multi-view stereo techniques such as [9] or [11]. Another option would be to use better reference objects, but the good results obtained in [5] indicate that

the linear combination model need not perfectly match the target material. We also believe that better results could be obtained using more highly-textured surfaces, because voxels not on the surface are more clearly inconsistent for highly-textured surfaces.

Previous work using orientation-consistency [5] yielded results with more detail and less noise than ours. We believe the explanation is that the results in [5] were run at a higher resolution (one normal per pixel) and required estimating fewer parameters. A key advantage of our work, however, is that we create full object models as opposed to a single depth map. While, in principle, multiple depth maps created from photometric stereo methods such as [5] could be merged together into a full object model, we expect that global distortions incurred from normal integration errors would make such a merging procedure difficult. An additional advantage over [5] is that we properly account for cast shadows.

In general, this work suggests that bridging the gap between photometric techniques such as orientation-consistency and multiview techniques such as voxel coloring is a promising avenue in geometric reconstruction. We have found that recovering normals yields much finer surface detail than is possible with stereo methods alone, but an important open problem is finding better ways of merging geometric and photometric constraints on shape.

**Acknowledgements.** This work was supported in part by NSF grants IIS-0049095 and IIS-0113007, an ONR YIP award, a grant from Microsoft Corporation, the UW Animation Research Labs, an NSERC Discovery Grant, the Connaught Fund, and an NSF Graduate Research Fellowship. Portions of this work were performed while Aaron Hertzmann was at the University of Washington.

## References

1. Carceroni, R.L., Kutulakos, K.N.: Multi-view scene capture by surfel sampling: From video streams to non-rigid 3d motion, shape and reflectance. International Journal of Computer Vision **49** (2002) 175–214
2. Jin, H., Soatto, S., Yezzi, A.: Multi-view stereo beyond Lambert. In: Proceedings of the 9th International Conference on Computer Vision. (2003)
3. Yang, R., Pollefeys, M., Welch, G.: Dealing with textureless regions and specular highlights–a progressive space carving scheme using a novel photo-consistency measure. In: Proceedings of the 9th International Conference on Computer Vision. (2003)
4. Zickler, T.E., Belhumeur, P.N., Kriegman, D.J.: Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. International Journal of Computer Vision **49** (2002) 215–227
5. Hertzmann, A., Seitz, S.M.: Shape and materials by example: A photometric stereo approach. In: Conference on Computer Vision and Pattern Recognition. (2003) 533–540
6. Seitz, S.M., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. International Journal of Computer Vision **35** (1999) 151–173

7. Kutulakos, K.N., Seitz, S.M.: A theory of shape by space carving. International Journal of Computer Vision **38** (2000) 199–218
8. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: 7th European Conference on Computer Vision. (2002) 82–96
9. Broadhurst, A., Drummond, T., Cipolla, R.: A probabilistic framework for space carving. In: Proceedings of the 8th International Conference on Computer Vision. (2001) 388–393
10. Slabaugh, G.G., Culbertson, W.B., Malzbender, T., Stevens, M.R., Schafer, R.W.: Methods for volumetric reconstruction of visual scenes. International Journal of Computer Vision **57** (2004) 179–199
11. Faugeras, O.D., Keriven, R.: Complete dense stereovision using level set methods. In: 5th European Conference on Computer Vision (1998). (1998) 379–393
12. Zheng, J., Murata, A.: Acquiring a complete 3D model from specular motion under the illumination of circular-shaped light sources. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 913–920
13. Bonfort, T., Sturm, P.: Voxel carving for specular surfaces. In: Proceedings of the 9th International Conference on Computer Vision. (2003)
14. Savarese, S., Perona, P.: Local analysis for 3d reconstruction of specular surfaces - part ii. In: 7th European Conference on Computer Vision. (2002) 759–774
15. Silver, W.M.: Determining shape and reflectance using multiple images. Master's thesis, MIT, Cambridge, MA (1980)
16. Woodham, R.J.: Photometric method for determining surface orientation from multiple images. Optical Engineering **19** (1980) 139–144
17. Wolff, L.B., Shafer, S.A., Healey, G.E., eds.: Physics-based vision: Principles and Practice, Shape Recovery. Jones and Bartlett, Boston, MA (1992)
18. Savarese, S., Rushmeier, H., Bernardini, F., Perona, P.: Shadow carving. In: Proceedings of the 8th International Conference on Computer Vision. (2001)
19. Blinn, J.F.: Simulation of wrinkled surfaces. In: Computer Graphics (Proceedings of SIGGRAPH). Volume 12. (1978) 286–292
20. Bouguet, J.Y.: Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/` (2004)

# Adaptive Probabilistic Visual Tracking with Incremental Subspace Update

David Ross[1], Jongwoo Lim[2], and Ming-Hsuan Yang[3]

[1] University of Toronto, Toronto, ON M5S 3G4, Canada
[2] University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA
[3] Honda Research Institute, Mountain View, CA 94041, USA
dross@cs.toronto.edu jlim1@uiuc.edu myang@honda-ri.com

**Abstract.** Visual tracking, in essence, deals with non-stationary data streams that change over time. While most existing algorithms are able to track objects well in controlled environments, they usually fail if there is a significant change in object appearance or surrounding illumination. The reason being that these visual tracking algorithms operate on the premise that the models of the objects being tracked are invariant to internal appearance change or external variation such as lighting or viewpoint. Consequently most tracking algorithms do not update the models once they are built or learned at the outset. In this paper, we present an adaptive probabilistic tracking algorithm that updates the models using an incremental update of eigenbasis. To track objects in two views, we use an effective probabilistic method for sampling affine motion parameters with priors and predicting its location with a maximum *a posteriori* estimate. Borne out by experiments, we demonstrate the proposed method is able to track objects well under large lighting, pose and scale variation with close to real-time performance.

## 1 Introduction

Visual tracking essentially deals with non-stationary data, both the object and the background, that change over time. Most existing algorithms are able to track objects, either previously viewed or not, in a short span of time and in a well controlled environment. However these algorithms usually fail to observe the object motion or have significant drifts after some period of time, either due to the drastic change of the object appearance or large lighting variation in the surroundings. Although such situations can be ameliorated with recourse to view-based appearance models [1] [2], adaptive color-based trackers [3] [4], contour-based trackers [5] [4], particle filters [5], 3D model based methods [6], optimization methods [1] [7], and background modeling [8], most algorithms typically operate on the premise that the target object models do not change drastically over time. Consequently these algorithms build or learn models of the objects first and then use them for tracking, without adapting the models to account for changes of the appearance of the object, e.g., large variation of pose or facial expression, or the surroundings, e.g., lighting variation. Such

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 470–482, 2004.
© Springer-Verlag Berlin Heidelberg 2004

an approach, in our view, is prone to performance instability and needs to be addressed for building a robust tracker.

In this paper, we present an efficient and adaptive algorithm that incrementally updates the model of the object being tracked. Instead of using a simple contour to enclose an image region or color pixels to represent moving "stuff" [9] for target tracking, we use an eigenbasis to represent the "thing" being tracked. In addition to providing a compact representation of the model based on the reconstruction principle, the eigenbasis approach also renders a probabilistic interpretation and facilitates efficient computation. Furthermore, it has been shown that for a Lambertian object the set of all images taken under all lighting conditions forms a convex polyhedral cone in the image space [10], and this polyhedral cone can be approximated well by a low-dimensional linear subspace using an eigenbasis [11] [12].

Given an observed image, we track the object based on a maximum *a posteriori* estimate of location - the image region near the predicted position that can be best approximated by the current eigenbasis. We compute this estimate using samples of affine parameters, describing the frame to frame motion of the target, drawn from their prior distributions, and combining them with the likelihood of the observed image region under our model.

The remaining part of this paper is organized as follows. We first review the most relevant tracking work in the next section. The details of our tracking algorithm is presented in Section 3, followed by numerous experiments to demonstrate its robustness under large pose and lighting variation. We conclude this paper with remarks on possible extensions for future work.

## 2 Context and Previous Work

There is an abundance of visual tracking work in the literature, from a simple two-view template matching approach [13] to a 3D model-based algorithm [6]. These algorithms differ mainly in the representation scheme – ranging from color pixels, blobs, texture, features, image patches, templates, active contours, snakes, wavelets, eigenspace, to 3D geometric models – and in the prediction approach, such as correlation, sum of square distance, particle filter, Kalman filter, EM algorithm, Bayesian inference, statistical models, mixture models, and optimization formulations. A thorough discussion of this topic is beyond the scope of this paper, thus in this section we review only the most relevant object tracking work and focus on the algorithms that operate directly on gray scale images.

In [1] Black and Jepson advocated a view-based eigenbasis representation for object tracking and formulated the two-view matching process as an optimization problem. Black et al. later extended the eigentracking algorithm to a mixture model to account for changes in object appearance [14]. The major advantages of using an eigenbasis representation are that it allows the tracker to have the notion of the "thing" being tracked, and the tracking algorithm operates on the subspace constancy assumption as opposed to the brightness constancy assumption of optical flow estimation. One disadvantage of the abovementioned algorithms is the use of viewed-based representation. In other words, one needs to learn the

eigenbasis of an object at each viewpoint before tracking, and these subspaces do not adapt over time. Furthermore, these algorithms need to solve a complex optimization problem [1], perhaps by using an iterative EM algorithm [14].

Brand proposed a method to track and build 3D models from video [15] [16] in which a set of feature points are manually selected and tracked, thereby obtaining 3D models based on the structure from motion principle. An incremental update of SVD, akin to [17], was introduced and improved to handle missing data. However, this method performs well only when the object is close to the camera with limited object motion.

Birchfield used a simple ellipse to enclose the region of interest, and integrated color and gradient information to track human heads [3]. This algorithm was further extended by Wu and Huang [4] in which they formulated the problem using a graphical model with a sequential Monte Carlo sampling method for estimating the state variables and thereby the object location. Their sampling method is largely based on the Condensation algorithm [5]. Although these methods perform well in constrained environments, the representation scheme is rather primitive, e.g., color and contour, in order to reduce the size of state space, and the tracker simply treats the target region as moving "stuff," paying little attention to what lies inside the contour. In other words, such trackers do not adapt to appearance change of the object and are likely to fail under large illumination change. Comaniciu and Meer presented the mean-shift algorithm for estimating how the mode of a density function changes over time [18], and then applied it to object tracking [19] using the histogram of color pixels. Due to the use of a simple pixel-based representation, it is not clear whether this algorithm will perform well under large illumination change.

A few attempts have been proposed to track objects under large illumination change [20] [6] [14]. These algorithms follow the same line of work of [21] and use a low dimensional linear subspace to approximate the space of all possible images of the object under different lighting conditions. Though they have demonstrated good empirical results, one needs to construct the basis images from a set of images acquired at fixed pose under different lighting conditions before tracking.

Most recently, some attention has been paid to the development of tracking methods that adapt the object models to changes in the object appearance or surroundings [22] [23] [2]. In [22], De La Torre et al. developed a tracking algorithm based on [1] in which they built a specific eigenbasis for the person being tracked by performing singular value decomposition (SVD) on a subset of training images which were most similar to the incoming test frames. Skin color was used to segment a face from the background and the affine motion parameters were estimated using a Kalman filter. Jepson et al. proposed the $\mathcal{WSL}$ model for learning adaptive appearance models in the context of object tracking [23]. They used the response of wavelet filters for object representation, and a mixture model to handle possible tracking scenarios. The weights in the mixture model are estimated using the EM algorithm and the affine motion parameters are computed using the stable features from wavelet filters. While their adaptive model is able to handle appearance and lighting change, the authors pointed out that it is possible for their model to learn the stable structure of the background if the background moves consistently with the foreground object over a period of

time. Consequently, their model may drift from the target object and lose track of it.

Our approach bears some resemblance to the classic particle filter algorithms [5], but with a more informative representation through the use of an eigenbasis. On the other hand, our approach is significantly different from the eigentracking approach [1]. First, we constantly update the eigenbasis using an computationally efficient algorithm. Consequently, our tracker is able to follow objects under large lighting and pose variation without recourse *a priori* to the illumination cone algorithm [24] or view-based approaches [1] . Second, we use a sampling technique to predict the object location without solving computationally expensive complex optimization problems. Our current implementation in MATLAB runs about 8 frames per second on a standard computer, and can certainly be improved to operate in real time. Third, our sampling technique can be extended, akin to the Condensation algorithm [5], to predict the location based on sequential observations, incorporating multiple likely hypotheses. Finally, the adaptive eigenbasis approach facilitates object recognition, thereby solving tracking and recognition problem simultaneously.

## 3 Adaptive Probabilistic Tracking

We detail the proposed algorithm and contrast the differences between this work and prior art in this section. Possible extensions of our algorithm are also discussed in due context.

### 3.1 Probabilistic Model

The tracking algorithm we propose is cast as an inference problem in a probabilistic Markov model, similar to the Hidden Markov Model and Kalman Filter [25]. At each time step $t$ we observe an image region $F_t$ of the sequence, and the location of the target object, $L_t$, is treated as an unobserved state variable. The motion of the object from one frame to the next is modeled by a distribution $p(L_t|L_{t-1})$, indicating the probability of the object appearing at $L_t$, given that it was just at $L_{t-1}$. This distribution encodes our beliefs about where the object might be at time $t$, prior to observing the current image region. Given $F_t$, we model the likelihood that the object is located at $L_t$ with the distribution $p(F_t|L_t)$. Using Bayes' rule to incorporate our observation with our prior belief, we conclude that the most probable *a posteriori* object location is at the maximum $l_t^*$ of $p(L_t|F_t, L_{t-1}) \propto p(F_t|L_t)p(L_t|L_{t-1})$. A graphical depiction of this model is illustrated in Figure 1.

We represent $L_t$, the location of the object at time $t$, using the four parameters of a similarity transformation ($x_t$ and $y_t$ for translation in $x$ and $y$, $r_t$ for rotation, and $s_t$ for scaling). This transformation warps the image, placing the target window - the object being tracking - in a rectangle centered at coordinates (0,0), with the appropriate width and height. This warping operates as a function of an image region $F$ and the object location $L$, i.e., $w(F, L)$.

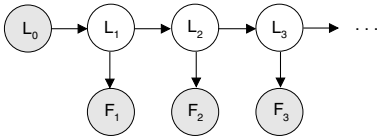



**Fig. 1.** Graphical model of the proposed tracking algorithm.

Our initial prior over locations assumes that each parameter is independently distributed, according to a normal distribution, around a predetermined location $L_0$. Specifically

$$p(L_1|L_0) = N(x_1; x_0, \sigma_x^2)N(y_1; y_0, \sigma_y^2)N(r_1; r_0, \sigma_r^2)N(s_1; s_0, \sigma_s^2) \tag{1}$$

where $N(z; \mu, \sigma^2)$ denotes evaluation of the normal distribution function for data point $z$, using the mean $\mu$ and variance $\sigma^2$.

Since our aim is to use an eigenbasis to model object appearance, we employ a probabilistic principal components distribution [26] (also known sensible PCA [27]) to model our image observation process. Given a location $L_t$, we assume the observed image region was generated by sampling an appearance of the object from the eigenbasis, and inserting it at $L_t$. Following Roweis [27], the probability of observing a datum $z$ given the eigenbasis $B$ and mean $\mu$ is $N(z; \mu, BB^T + \varepsilon I)$, where the $\varepsilon I$ term corresponds to the covariance of additive Gaussian noise present in the observation process. In the limit as $\varepsilon \to 0$, $N(z; \mu, BB^T + \varepsilon I)$ is proportional to the negative exponential of the squared distance between $z$ and the linear subspace $B$, $\|(z - \mu) - BB^T(z - \mu)\|^2$.

### 3.2 Predicting Object Location

According to our probabilistic model, since $L_t$ is never directly observed, full Bayesian inference would require us to compute the distribution $P(L_t|\ F_t, F_{t-1}, \ldots, F_t, L_0)$ at each time step. Unfortunately this distribution is infeasible to compute in closed form. Instead, we will approximate it using a normal distribution of the same form as our prior in Equation 1 around the maximum $l_t^*$ of $p(L_t|F_t, l_{t-1}^*)$.

We can efficiently and effectively compute an approximation to $l_t^*$ using a simple sampling method. Specifically, we begin by drawing a number of sample locations from our prior $p(L_t|l_{t-1}^*)$. For each sample $l_s$ we compute its posterior probability $p_s = p(l_s|F_t, l_{t-1}^*)$. The posterior $p_s$ is simply the likelihood of $l_s$ under our probabilistic PCA distribution, times the probability with which $l_s$ was sampled, disregarding the normalization factor which is constant across all samples. Finally we select the sample with the largest posterior to be our approximate $l_t^*$, i.e.,

$$l_t^* = \arg\max_{l_s} p(l_s|F_t, l_{t-1}^*) \tag{2}$$

This method has the nice property that a single parameter, namely the number of samples, can be used to control the trade-off between speed and tracking accuracy.

To allow for incremental updates to our object model, we specifically do not assume that the probability distribution of observations remains fixed over time. Rather, we use recent observations to update this distribution, albeit in a non-Bayesian fashion. Given an initial eigenbasis $B_{t-1}$, and a new appearance $w_{t-1} = w(F_{t-1}, l^*_{t-1})$ we compute a new basis $B_t$ using the sequential Karhunen-Loeve (K-L)algorithm [17] (see next section). The new basis is used when calculating $p(F_t|L_t)$. It is also possible to perform an on-line update of the mean of the probabilistic PCA model.

The proposed sampling method is flexible and can be applied to localize targets in the first frame, though manual initialization or sophisticated object detection algorithms are applicable. By specifying a broad prior (perhaps uniform) over the entire image, and drawing enough samples, our tracker could locate the target by the maximum response using the current distribution and the initial eigenbasis.

Finally the parametric sampling method in the current implementation can be extended, akin to particle filters such as the Condensation algorithm [5], to integrate temporal information by a recursive Bayesian formulation, and allow multiple hypotheses by using a non-parametric distribution.

### 3.3 Incremental Update of Eigenbasis

Since visual tracking deals with time-varying data, and we use an eigenbasis for object representation, it is imperative to continually update the eigenbasis from the time-varying covariance matrix. This problem has been studied in the signal processing community, where several computationally efficient techniques have been proposed in the form of recursive algorithms [28]. In this paper, we use a variant of the efficient sequential Karhunen-Loeve algorithm to update the eigenbasis [17], which in turns is based on the classic R-SVD method [29].

Let $X = U\Sigma V^T$ be the SVD of a data $M \times P$ matrix $X$ where each column vector is an observation (e.g., image). The R-SVD algorithm provides an efficient way to carry out the SVD of a larger matrix $X^* = (X|E)$, where $E$ is a $M \times K$ matrix consisting of $K$ additional observations (e.g., incoming images) as follows.

- Use an orthonormaliztion process (e.g., Gram-Schmidt algorithm) on $(U|E)$ to obtain an orthonormal matrix $U' = (U|\tilde{E})$.
- Form the matrix $V' = \begin{pmatrix} V & 0 \\ 0 & I_K \end{pmatrix}$, where $I_K$ is a $K$ dimensional identity matrix.
- Let $\Sigma' = U'^T X^* V' = \begin{pmatrix} U^T \\ \tilde{E}^T \end{pmatrix} (X|E) \begin{pmatrix} V & 0 \\ 0 & I_K \end{pmatrix} = \begin{pmatrix} U^T XV & U^T E \\ \tilde{E}^T XV & \tilde{E}^T E \end{pmatrix} = \begin{pmatrix} \Sigma & U^T E \\ 0 & \tilde{E}^T E \end{pmatrix}$ since $\Sigma = U^T XV$ and $\tilde{E}^T XV = 0$. Notice that the $K$ rightmost columns of $\Sigma'$ are the new vectors (images), represented in the updated orthonormal basis spanned by the columns of $U'$.
- Compute the SVD of $\Sigma' = \tilde{U}\tilde{\Sigma}\tilde{V}^T$ and the SVD of $X^*$ is

$$X^* = U'(\tilde{U}\tilde{\Sigma}\tilde{V}^T)V'^T = (U'\tilde{U})\tilde{\Sigma}(\tilde{V}^T V'^T) \tag{3}$$

By exploiting the orthonormal properties and block structure, the SVD computation of $X^*$ can be efficiently carried by using the smaller matrices, $U'$, $V'$,

$\Sigma'$ and the SVD of smaller matrix $\Sigma'$. The computational complexity analysis and details of the R-SVD algorithm are described in [29].

Based on the R-SVD method, the sequential Karhunen-Loeve algorithm further exploits the low dimensional subspace approximation and only retains a small number of eigenvectors as new data arrive. See [17] for details of an update strategy and the computational complexity analysis.

### 3.4 Proposed Tracking Algorithm

Our tracking algorithm is flexible in the sense that it can be carried out with or without an initial eigenbasis of the object. For the case where training images of the object are available and well cropped, an eigenbasis can be constructed which the proposed tracker can use at the onset of tracking. However situations arise where we do not have training images at our disposal. In such cases, the tracker can gradually construct and update an eigenbasis from incoming images if the object is localized in the first frame. An application of this technique is demonstrated experimentally in the following section.

Putting the inference, sampling and subspace update modules together, we obtain an adaptive probabilistic tracking algorithm as follows:

1. **(Optional) Construct an initial eigenbasis:** From a set of training images of the object, or similar objects, learn an initial eigenbasis. This includes the following steps: a) histogram-equalize all of the training images b) subtract the mean from the data c) compute the desired number principle components.
2. **Choose the initial location:** Given the tracking sequence, locate the object in the first frame. This can be done manually, or by using an automatic object detector. Alternatively, we can draw more samples to detect objects in the first frame if an eigenbasis of the object is available.
3. **Search possible locations:** Draw a number of samples from the prior distribution over locations. For each location, obtain the image region, histogram equalize it, subtract the mean learned during the training phase, and compute it's probability under the current eigenbasis.
4. **Predict the most likely location:** Select the location with the maximum *a posteriori* probability. Update the prior to be centered at this location.
5. **Update the eigenbasis:** Using the image window selected in 3, update the eigenbasis using the sequential Karhunen-Loeve algorithm. Note that this need not be performed for every frame. Instead, it is possible and possibly preferable to store the tracked image regions for a number of previous frames and perform a batch update.
6. Go to Step 3.

For most vision applications, it suffices to use a small number of eigenvectors (those with the largest eigenvalues). To achieve this, one may simply discard the unwanted eigenvectors from the initial basis (Step 1c), and from the updated basis after each invocation of sequential K-L (Step 5).

As a result of keeping only the top eigenvectors at each stage of the sequential SVD (equation 3), the samples seen at an earlier time are gradually forgotten.

None of the tracking experiments shown in the next section are improved by using an explicit forgetting factor, but it is likely that an explicit forgetting factor helps in certain situations.

In this work, the mean of the subspace is not updated. Any changes in the mean can be compensated for, without loss of modeling accuracy, by simply including additional eigenvector in the basis [30]. In fact, some studies have shown that uncentered PCA (in which the mean is not subtracted) performs as well as centered PCA [30]. Empirical results (see next section) show that our method with incremental update performs well under large variation in lighting, pose and deformations. Our future work will focus on developing a method to update the subspace with running means.

## 4 Experiments

We conducted numerous experiments to test whether the proposed tracking algorithm performs well in terms of following the object position and updating the appearance model. All the image sequences consist of $320 \times 240$ pixel grayscale videos, recorded at 30 frames/second and 256 gray-levels per pixel. As a baseline, we compared our algorithm with three other trackers. The first is an eigentracker using a fixed basis, which was was implemented by removing the sequential K-L updates from our tracker. The second is a simple template tracker that, at each time step, searches for the window most like the appearance of the object in the first frame of the sequence. The third one is a two-view tracker, which searches at each time step for the window most resembling the appearance in the preceding frame based on sum of square errors.

For all the experiments, the parameters of the prior distribution of affine parameters were, $\sigma_x = \sigma_y = 5$, $\sigma_r = \sigma_s = 0.1$. Typically we used between 100 and 500 samples, but good results can be obtained using as few as 64. For the experiments with face objects, we built an initial eigenbasis using a set of 2901 well-cropped $19 \times 19$ face images from the MIT CBCL data set[1]. We restricted this eigenbasis to the top 50 eigenvectors. For efficiency, an eigenbasis is updated every 5 frames using the sequential Karhunen-Loeve algorithm.

Figure 2 shows the empirical results where the tracked object is enclosed by a rectangle by our algorithm and the top 10 eigenvectors at each time instance are shown. The object location in the first frame is manually initialized and and the eigenbasis is built from the MIT face data set. Notice that in the first few frames, the eigenvectors resemble the generic object patterns (upper left panel). Since the eigenbasis is constantly updated with the incoming frames of the same object, the eigenbasis gradually shifts to capture the details of that target as shown by the eigenvectors in the following two panels (e.g., the eye glasses are visible in the first eigenvectors of the upper middle and upper right panels) though a few eigenvectors still contain high frequency noise. As this target changes its pose, the updated subspace also accounts for the variation in appearance (see the first three eigenvectors in upper middle panel). Finally with

[1] CBCL Face Database #1, MIT Center For Biological and Computation Learning, available at `http://www.ai.mit.edu/projects/cbcl`

**Fig. 2.** Tracking an object undergoing a large pose variation (See video at `http://www.cs.toronto.edu/~dross/ivt/`).

enough samples, the subspace has been updated that capture more facial details of this individual (lower three panels). Notice that the proposed algorithm tracks the target undergoing large pose variation. On the other hand, the eigentracker with a fixed eigenbasis fails as this person changes pose. Though the eigentracker can be further improved with view-based subspaces, it is difficult to collect the training images for all possible pose variations. We note that all the other three trackers mentioned above fail to track the target object after a short period of time, especially when there is a large pose variation. The fixed template tracker fails during the first out-of-plane rotation of the subject's face. In this video, the two-view tracker quickly drifts away from the subject's face, instead tracking the side of his head.

Our algorithm is able to track and learn the eigenbasis of the target object under large appearance and lighting variation, as demonstrated in Figure 3. The target object is first manually located, and the initial eigenbasis is again learned from the MIT data set. Notice that the eigenbasis adapts from a generic eigenbasis to one that captures the details of this individual as shown evidently in the first few eigenvectors of all the panels. The third eigenvector of the upper middle panel captures the facial expression variation. Notice also that there is drastic lighting change that changes the facial appearance in the second, third, and fourth panels. Our tracker is still able to follow the target object in these situations. On the other hand, both the eigentracker and template tracker fail to

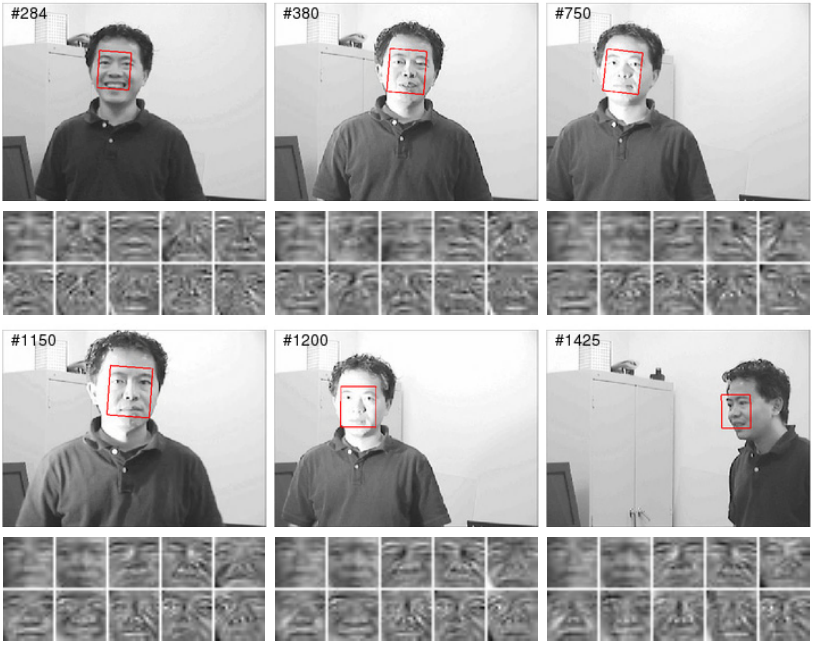



**Fig. 3.** Tracking an object undergoing a large appearance, pose, and lighting variation (See video at `http://www.cs.toronto.edu/~dross/ivt/`).

track the object when there is large lighting variation (starting from the middle panel). The two-view tracker suffers during the scale change in this sequence, drifting to track a small subregion of the face as the subject approaches the camera. Finally, the proposed algorithm is able to track the target object under a combination of lighting, expression and pose variation (lower right panel).

Figure 4 shows an example where we do not have an eigenbasis of the target object to begin with. In such situations, our algorithm is still able to gradually learn an eigenbasis and use that for tracking. The upper left panel shows the first few eigenvectors at the 6th frame of the sequence, immediately following the first eigenbasis update. Note that the first eigenvector captures some details of the target object with only a few frames, while the remaining few eigenvectors encode high frequency noise. As more frames become available, our method clearly learns an eigenbasis of the object, as shown in the first few eigenvectors of the top middle panel. As the object moves, our algorithm uses the learned eigenbasis for tracking and continues to update the appearance model. Notice that the tracker is able to follow the object well when it undergoes large pose and scale variation (See video at `http://www.cs.toronto.edu/~dross/ivt/`).

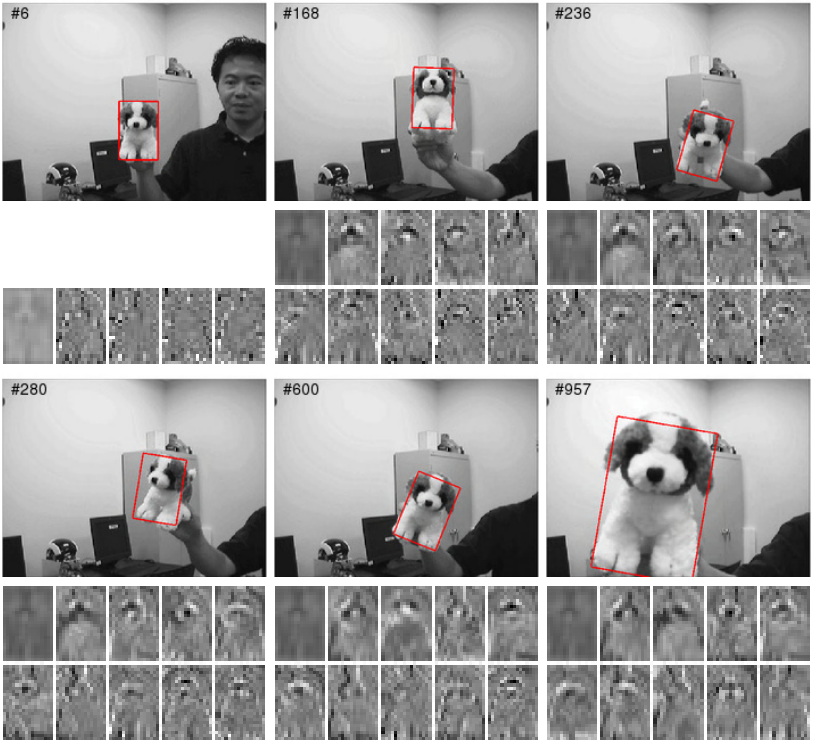



**Fig. 4.** Learning an eigenbasis while tracking the target object.

## 5 Concluding Remarks and Future Work

We have presented an adaptive probabilistic visual tracking algorithm that constantly updates its appearance model to account for the intrinsic (e.g. facial expression and pose) and extrinsic variation (e.g. lighting). To track an object in two views, we use an effective probabilistic method for sampling affine motion parameters with priors and predict its location with maximum a posteriori estimate. Through experiments, we demonstrated that the proposed method is able to track objects well in real-time under large lighting, pose and scale variation.

Though the proposed algorithm works well under large appearance, pose, lighting, and scale variation, our tracker sometimes drifts off by a few pixels before recovering in later frames. In addition, the current tracker does not handle occlusion well. This problem can be ameliorated by computing a mask before carrying out the eigen-decomposition [1] or using a method to deal with missing data [15], and will be addressed in our future work. A more selective adaption mechanism, e.g. [31], is also required to ensure the integrity of the newly arrived samples before updating the eigenbasis. Meanwhile, we plan to investigate better sampling strategy and extend our work to integrate temporal information. Furthermore, we plan to extend the current work to use eigenbases for constructing illumination cone for fixed pose, and for object recognition.

**Acknowledgements.** This work was carried out while the first two authors visited at Honda Research Institute in the summer of 2003. We appreciate the valuable comments and suggestions of the anonymous reviewers. We thank the MIT Center For Biological and Computation Learning for providing data used in our experiments (CBCL Face Database #1 - `http://www.ai.mit.edu/projects/cbcl`).

## References

1. Black, M.J., Jepson, A.D.: Eigentracking: Robust matching and tracking of articulated objects using view-based representation. In Buxton, B., Cipolla, R., eds.: Proceedings of the Fourth European Conference on Computer Vision. LNCS 1064, Springer Verlag (1996) 329–342
2. Morency, L.P., Rahimi, A., Darrell, T.: Adaptive view-based appearance models. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2003) 803–810
3. Birchfield, S.: Elliptical head tracking using intensity gradient and color histograms. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1998) 232–37
4. Wu, Y., Huang, T.: A co-inference approach for robust visual tracking. In: Proceedings of the Eighth IEEE International Conference on Computer Vision. Volume 2. (2001) 26–33
5. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In Buxton, B., Cipolla, R., eds.: Proceedings of the Fourth European Conference on Computer Vision. LNCS 1064, Springer Verlag (1996) 343–356
6. La Cascia, M., Sclaroff, S., Athitsos, V.: Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3D models. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 322–336
7. Black, M.J., Fleet, D.J., Yacoob, Y.: Robustly estimating changes in image appearance. Computer Vision and Image Understanding **78** (2000) 8–31
8. Harville, M.: A framework for high-level feedback to adaptive, per-pixel mixture of Gaussian background models. In Heyden, A., Sparr, G., Nielsen, M., Johansen, P., eds.: Proceedings of the Seventh European Conference on Computer Vision. LNCS 2352, Springer Verlag (2002) 531–542
9. Adelson, E.H., Bergen, J.R.: The plenoptic function and the elements of early vision. In Landy, M., Movshon, J.A., eds.: Computational Models of Visual Processing. MIT Press (1991) 1–20
10. Belhumeur, P.N., Kriegman, D.J.: What is the set of images of an object under all possible illumination conditions. International Journal of Computer Vision **28** (1998) 1–16
11. Hallinan, P.: A low-dimensional representation of human faces for arbitrary lighting conditions. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1994) 995–999
12. Basri, R., Jacobs, D.: Lambertian reflectance and linear subspaces. In: Proceedings of the Eighth IEEE International Conference on Computer Vision. Volume 2. (2001) 383–390
13. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of International Joint Conference on Artificial Intelligence. (1981) 674–679

14. Black, M.J., Fleet, D.J., Yacoob, Y.: A framework for modeling appearance change in image sequence. In: Proceedings of the Sixth IEEE International Conference on Computer Vision. (1998) 660–667
15. Brand, M.: Morphable 3D models from video. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2001) 315–322
16. Brand, M.: Incremental singular value decomposition of uncertain data with missing values. In Heyden, A., Sparr, G., Nielsen, M., Johansen, P., eds.: Proceedings of the Seventh European Conference on Computer Vision. LNCS 2350, Springer Verlag (2002) 707–720
17. Levy, A., Lindenbaum, M.: Sequential Karhunen-Loeve basis extraction and its application to images. IEEE Transactions on Image Processing **9** (2000) 1371–1374
18. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 603–619
19. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003) 564–577
20. Hager, G., Belhumeur, P.: Real-time tracking of image regions with changes in geometry and illumination. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (1996) 403–410
21. Shashua, A.: Geometry and Photometry in 3D Visual Recognition. PhD thesis, Massachusetts Institute of Technology (1992)
22. De la Torre, F., Gong, S., McKenna, S.J.: View-based adaptive affine tracking. In Burkhardt, H., Neumann, B., eds.: Proceedings of the Fifth European Conference on Computer Vision. LNCS 1406, Springer Verlag (1998) 828–842
23. Jepson, A.D., Fleet, D.J., El-Maraghi, T.F.: Robust online appearance models for visual tracking. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 1. (2001) 415–422
24. Hager, G.D., Belhumeur, P.N.: Efficient region tracking with parametric models of geometry and illumination. IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 1025–1039
25. Jordan, M.I., ed.: Learning in Graphical Models. MIT Press (1999)
26. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society, Series B **61** (1999) 611–622
27. Roweis, S.: EM algorithms for PCA and SPCA. In Jordan, M.I., Kearns, M.J., Solla, S.A., eds.: Advances in Neural Information Processing Systems 10, MIT Press (1997) 626–632
28. Champagne, B., Liu, Q.G.: Plane rotation-based EVD updating schemes for efficient subspace tracking. IEEE Transactions on Signal Processing **46** (1998) 1886–1900
29. Golub, G.H., Van Loan, C.F.: Matrix Computations. The Johns Hopkins University Press (1996)
30. Jolliffe, I.T.: Principal Component Analysis. Springer-Verlag (2002)
31. Vermaak, J., Pérez, P., Gangnet, M., Blake, A.: A framework for high-level feedback to adaptive, per-pixel mixture of Gaussian background models. In Heyden, A., Sparr, G., Nielsen, M., Johansen, P., eds.: Proceedings of the Seventh European Conference on Computer Vision. LNCS 2350, Springer Verlag (2002) 645–660

# On Refractive Optical Flow

Sameer Agarwal, Satya P. Mallick, David Kriegman, and Serge Belongie

University of California, San Diego, La Jolla CA 92093, USA,
{sagarwal@cs,spmallick@graphics,kriegman@cs,sjb@cs}.ucsd.edu,
http://vision.ucsd.edu/

**Abstract.** This paper presents a novel generalization of the optical flow equation to the case of refraction, and it describes a method for recovering the refractive structure of an object from a video sequence acquired as the background behind the refracting object moves. By structure here we mean a representation of how the object warps and attenuates (or amplifies) the light passing through it. We distinguish between the cases when the background motion is known and unknown. We show that when the motion is unknown, the refractive structure can only be estimated up to a six-parameter family of solutions without additional sources of information. Methods for solving for the refractive structure are described in both cases. The performance of the algorithm is demonstrated on real data, and results of applying the estimated refractive structure to the task of environment matting and compositing are presented.

## 1 Introduction

The human visual system is remarkable in its ability to look at a scene through a transparent refracting object and to deduce the structural properties of that object. For example, when cleaning a wine glass, imperfections or moisture may not be visible at first, but they become apparent when one holds the glass up and moves or rotates it. We believe that the primary cue here is the optical flow of the background image as observed through the refracting object, and our aims in this paper are to build a theory of how motion can be used for recovering the structure of a refracting object, to introduce algorithms for estimating this structure, and to empirically validate these algorithms. By structure here we mean a representation of how the object warps and attenuates (or amplifies) the light passing through it. Recall, as a light ray enters or exits the object, its direction changes according to Snell's Law (known as Descartes' Law in France). Furthermore, the emitted radiance may differ from the incident radiance due to the difference in solid angle caused by the geometry of the interfaces between the object and the air as well as absorption along the light ray's path through the object. The geometric shape of the object itself, while of independent interest, is not the focus of our inquiry here.

The primary contribution of this paper is to generalize the optical flow equation to account for the warping and attenuation caused by a refractive object, and to present algorithms for solving for the warp and attenuation using a sequence of images obtained as a planar background moves behind the refracting

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 483–494, 2004.
© Springer-Verlag Berlin Heidelberg 2004

object. Both the case where the background motion is known and where it is unknown are considered. We demonstrate the performance of these algorithms on both synthetic and real data.

While there is a vast literature on motion understanding including transparency, the warp induced by refraction appears to have been neglected hitherto [1, 2,3,4]. The ability to recover the refractive structure of an object has a number of applications. Recently, environment matting and compositing have been introduced as techniques for rendering images of scenes which contains foreground objects that refract and reflect light [5,6,7]. The proposed approach offers a new method for creating a matte of real objects without the need for extensive apparatus besides a video camera. Refractive optical flow may also be useful for visual inspection of transparent/translucent objects.

This paper was inspired by the work of Zongker et. al [7] on environment matting and its subsequent extension by Chuang et. al. [5]. We discuss this method in the next section. However the work that comes closest to ours in spirit is that of H. Murase [8]. Murase uses optical flow to recover the shape of a flexible transparent object, e.g., water waves that change over time. To make the problem tractable he makes a number of simplifying assumptions (a) The camera is orthographic, (b) The refracting object is in contact with the background plane which has a flat shape and a static unknown pattern on it, (c) the average shape over time of the refracting surface is known *a priori* to have zero slope. In this paper our interest is in scenes where the refracting object is rigid and stationary. Beyond that, we make no assumptions about the number of objects, their refractive indices, or the positioning of the refracting object in the scene.We do not address effects due to specular, Fresnel or total internal reflections in this work.

The rest of the paper is organized as follows. In the next section we begin by describing our image formation model and we then introduce the notion of an optical kernel and describe how the choice of a particular optical kernel leads to a new generalization of the optical flow equation. Section 3 describes algorithms for solving for the refractive structure using this equation. Section 4 demonstrates the performance of our algorithm on synthetic and real data, and its application to matting. We conclude in Section 5 with a discussion and directions for future work.

## 2 A Theory of Refractive Optical Flow

In this section we describe our image formation model and use it to derive the refractive optical flow equation. As illustrated in Figure 1(a), we assume that the scene consists of three entities.

1. A background image plane $B$, where $B(\mathbf{u}, t)$ denotes the emitted radiance at the point $\mathbf{u} = (u, v)^\top$ and time $t$.
2. A foreground image plane $I$, where $I(\mathbf{x}, t)$ denotes the incident radiance at the point $\mathbf{x} = (x, y)^\top$ and time $t$.
3. A collection of one or more refracting objects between $I$ and $B$.

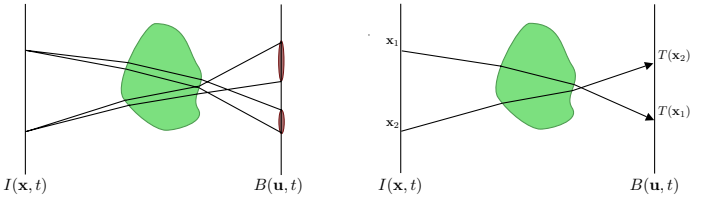



**Fig. 1.** The image formation model: (a) The general image formation model, where the incident irradiance at a point $\mathbf{x}$ in the foreground plane $I$ is a result of the emitted radiance of some number of points in the background plane $B$. (b) The single ray image formation model, where the incident irradiance at $\mathbf{x}$ is linear function of the emitted radiance at the point $T(\mathbf{x})$ in the background plane.

The illumination in the scene is assumed to be ambient or coming from a directional light source, and the background plane is assumed to be an isotropic radiator. The background and the foreground planes are not constrained to be parallel. No assumptions are made about the shape of the refracting objects, or optical properties of their constituent materials. We treat the refracting objects as a "black-box" function that warps and attenuates the light rays passing through it. It is our aim to recover this function from a small set of images of the scene. Given the assumptions about scene illumination stated earlier, the incident radiance at a point $\mathbf{x}$ in the foreground is a result of the reflected and emitted radiance from the background plane. Now let the function that indicates what fraction of the light intensity observed at a point $\mathbf{x}$ in the foreground comes from the point $\mathbf{u}$ in the background be denoted by the optical kernel $\mathcal{K}(\mathbf{u},\mathbf{x})$. The total light intensity at $\mathbf{x}$ can now be expressed as an integral over the entire background plane,

$$I(\mathbf{x},t) = \int \mathcal{K}(\mathbf{u},\mathbf{x}) B(\mathbf{u},t) d\mathbf{u} \tag{1}$$

The problem of recovering the refractive structure of an object can now be restated as the problem of estimating the optical kernel associated with it.

The set of all functions of the form $\mathcal{K}(\mathbf{u},\mathbf{x})$ is huge. A large part of this set consists of functions that violate laws of physics. However the set of physically plausible optical kernels is still very big, and the reconstruction of $\mathcal{K}$ using a small number of images is an ill-posed problem. Additional constraints must be used to make the problem tractable.

One such direction of inquiry is to assume a low dimensional form for $\mathcal{K}$ by a small set of parameters. Zongker et al. in their work on environment matting, assume a parametric box form for $\mathcal{K}(\mathbf{u},\mathbf{x})$,

$$\mathcal{K}(\mathbf{u},\mathbf{x}) = \begin{cases} 1/\mu(\mathbf{x}) & \text{if } a(\mathbf{x}) \le u \le b(\mathbf{x}) \,\&\, c(\mathbf{x}) \le v \le d(\mathbf{x}) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where, $a(\mathbf{x}), b(\mathbf{x}), c(\mathbf{x}), d(\mathbf{x})$ are functions of $\mathbf{x}$. $\mu(\mathbf{x})$ is the area of the rectangle enclosed by $(a, c)$ and $(b, d)$. The kernel maps the average intensity of a rectangular region in the background to a point in the foreground. The values of the parameters $\{a(\mathbf{x}), b(\mathbf{x}), c(\mathbf{x}), d(\mathbf{x})\}$ for each point $\mathbf{x}$ are determined using a set of calibrated background patterns and performing a combinatorial search on them so as to minimize the reconstruction error of the foreground image. Chuang et al. [5] generalize $\mathcal{K}$ to be from the class of oriented two-dimensional Gaussians. In both of these cases, knowledge of the background image is assumed.

In this paper we choose to pursue an alternate direction. We consider optical kernels of the form

$$\mathcal{K}(\mathbf{u}, \mathbf{x}) = \alpha(\mathbf{x})\delta(\mathbf{u} - T(\mathbf{x})) \tag{3}$$

where $\delta(\cdot)$ is Dirac's delta function, $T(\mathbf{x})$ is a piecewise differentiable function that serves as the parameter for the kernel indicating the position in the background plane where the kernel is placed when calculating the brightness at $\mathbf{x}$ in the foreground image plane. The function $\alpha(\mathbf{x})$ is a positive scalar function that accounts for the attenuation of light reaching $\mathbf{x}$. Figure 1(b) illustrates the setup.

In the following we will show how if we restrict ourselves to this class of optical kernels, we can recover the refractive structure without any knowledge of the background plane. We will also demonstrate with experiments how this subclass of kernels despite having a very simple description is capable of capturing refraction through a variety of objects.

The image formation equation can now be re-written as

$$I(\mathbf{x}, t) = \int \alpha(\mathbf{x})\delta(\mathbf{u} - T(\mathbf{x}))B(\mathbf{u}, t)d\mathbf{u} \tag{4}$$

$$= \alpha(\mathbf{x})B(T(\mathbf{x}), t) \tag{5}$$

We begin by differentiating the above equation w.r.t $\mathbf{x}$, to get

$$\nabla_{\mathbf{x}} I(\mathbf{x}, t) = (\nabla_{\mathbf{x}}\alpha(\mathbf{x}))B(T(\mathbf{x})) + \alpha(\mathbf{x})\mathbf{J}^{\top}(T(\mathbf{x}))(\nabla_{T(\mathbf{x})}B(T(\mathbf{x}), t)) \tag{6}$$

Here, $\mathbf{J}^{\top}(T(\mathbf{x}))$ is the transpose of the Jacobian of the transformation $T(\mathbf{x})$. Using Equation (5) the above equation can be written as

$$\nabla_{\mathbf{x}} I(\mathbf{x}, t) = I(\mathbf{x}, t)\frac{\nabla_{\mathbf{x}}\alpha(\mathbf{x})}{\alpha(\mathbf{x})} + \mathbf{J}^{\top}(T(\mathbf{x}))\left[\alpha(\mathbf{x})\nabla_{T(\mathbf{x})}B(T(\mathbf{x}))\right]$$

$$\mathbf{J}^{-\top}(T(\mathbf{x}))\left[\nabla_{\mathbf{x}} I(\mathbf{x}, t) - I(\mathbf{x}, t)\frac{\nabla_{\mathbf{x}}\alpha(\mathbf{x})}{\alpha(\mathbf{x})}\right] = \alpha(\mathbf{x})\nabla_{T(\mathbf{x})}B(T(\mathbf{x})) \tag{7}$$

Taking temporal derivatives of Equation (5) gives

$$I_t(\mathbf{x}, t) = \alpha(\mathbf{x})B_t(T(\mathbf{x}), t) \tag{8}$$

Now, let $\mathbf{c}(\mathbf{u}, t)$ denote the velocity of the point $\mathbf{u}$ at time $t$ in the background plane. Then from the Brightness Constancy Equation [9] we know

$$\mathbf{c}(\mathbf{u}, t)^{\top}\nabla_{\mathbf{u}}B(\mathbf{u}, t) + B_t(\mathbf{u}, t) = 0 \tag{9}$$

Now assuming that the background image undergoes in-plane translation with velocity $\mathbf{c}(\mathbf{u},t) = \mathbf{c}(T(\mathbf{x}),t)$, we take the dot product of Equation (7) with $\mathbf{c}(T(\mathbf{x}),t)$ and add it to Equation (8) to get

$$\mathbf{c}(T(\mathbf{x}),t)^\top \mathbf{J}^{-\top}(T(\mathbf{x}))\left[\nabla_{\mathbf{x}} I(\mathbf{x},t) - I(\mathbf{x},t)\frac{\nabla_{\mathbf{x}}\alpha(\mathbf{x})}{\alpha(\mathbf{x})}\right] + I_t(\mathbf{x},t) = \alpha(\mathbf{x})\left[\mathbf{c}(T(\mathbf{x}),t)^\top \nabla_{T(\mathbf{x})} B(T(\mathbf{x})) + B_t(T(\mathbf{x}))\right] \tag{10}$$

From Equation (9) we know that the right hand side goes to zero everywhere, giving us

$$\mathbf{c}(T(\mathbf{x}),t)^\top \mathbf{J}^{-\top}(T(\mathbf{x}))\left[\nabla_{\mathbf{x}} I(\mathbf{x},t) - I(\mathbf{x},t)\frac{\nabla_{\mathbf{x}}\alpha(\mathbf{x})}{\alpha(\mathbf{x})}\right] + I_t(\mathbf{x},t) = 0 \tag{11}$$

Now for simplification's sake, let $\beta(\mathbf{x}) = \log\alpha(\mathbf{x})$. Dropping the subscript on $\nabla$ we get

$$\boxed{\mathbf{c}(T(\mathbf{x}),t)^\top \mathbf{J}^{-\top}(T(\mathbf{x}))\left[\nabla I(\mathbf{x},t) - I(\mathbf{x},t)\nabla\beta(\mathbf{x})\right] + I_t(\mathbf{x},t) = 0} \tag{12}$$

This is the refractive optical flow equation.

### 2.1 Properties of the Refractive Optical Flow Equation

Before we dive into the solution of Equation (12) for recovering the refractive structure, we comment on its form and some of its properties. The first observation is that if there is no distortion or attenuation, i.e.,

$$T(\mathbf{x}) = \mathbf{x}, \qquad \alpha(\mathbf{x}) = 1$$

the Jacobian of $T$ reduces to the identity, and the gradient of $\beta(\mathbf{x})$ reduces to zero everywhere, giving us the familiar optical flow equation in the foreground image.

$$\mathbf{c}(\mathbf{x},t)^\top \nabla I(\mathbf{x},t) + I_t(\mathbf{x},t) = 0 \tag{13}$$

The second observation is that the equation is independent of $B(\mathbf{u},t)$. This means that we can solve for the refractive flow through an object just by observing a distorted version of the background image. Knowledge of the background image itself is not needed.

Finally we observe that Equation (12) is in terms of the Jacobian of $T$, i.e. any function $T'(\mathbf{x}) = T(\mathbf{x}) + \mathbf{u}_0$ will result in the same equation. This implies that $T$ can only be solved up to a translation ambiguity. Visually this is equivalent to viewing the scene through a periscope. The visual system has no way of discerning whether or not an image was taken through a periscope. A second ambiguity is introduced into the solution when we note that the velocity $\mathbf{c}(\mathbf{u},t)$ is in the background plane and there is nothing that constrains the two coordinate systems from having different scales along each axis. Hence $T(\mathbf{x})$ can only be recovered up to a four parameter family of ambiguities corresponding to translation and scaling. The attenuation function $\beta(\mathbf{x})$ is not affected by the scaling in $\mathbf{c}(\mathbf{u},t)$ and hence is recovered up to a translation factor which in turn means that $\alpha(\mathbf{x})$ is recovered up to a scale factor.

## 3 Solving the Equation of Refractive Flow

In this section, we further analyze Equation (12) for the purposes of solving it. We begin by considering a further simplification of Equation (12). We assume that the background plane translates with in-plane velocity $\mathbf{c}(\mathbf{u}, t) = \mathbf{c}(t)$ that is constant over the entire background plane. We consider two cases, the calibrated case (when the motion of the background $\mathbf{c}(t)$ is known) and the uncalibrated case (when the motion of the background is unknown). In each case we describe methods for recovering $T(\mathbf{x})$ and $\alpha(\mathbf{x})$, and note the ambiguities in the resulting solution. Let $T(\mathbf{x})$ be denoted by

$$T(x, y) = (g(x, y), h(x, y))^\top \tag{14}$$

The Jacobian of $T$ and its inverse transpose are

$$\mathbf{J}(T(\mathbf{x})) = \begin{bmatrix} g_x & g_y \\ h_x & h_y \end{bmatrix} \qquad \mathbf{J}^{-\top}(T(\mathbf{x})) = \frac{1}{g_x h_y - g_y h_x} \begin{bmatrix} h_y & -h_x \\ -g_y & g_x \end{bmatrix}$$

The translation velocity in the background plane is $\mathbf{c}(t) = (\xi(t), \eta(t))^\top$. Substituting these in Equation (12) we get

$$\begin{bmatrix} \xi & \eta \end{bmatrix} \frac{1}{g_x h_y - h_x g_y} \begin{bmatrix} h_y & -h_x \\ -g_y & g_x \end{bmatrix} \begin{bmatrix} I_x - \beta_x I \\ I_y - \beta_y I \end{bmatrix} + I_t = 0 \tag{15}$$

which rearranges to

$$\frac{\eta I_y g_x - \eta I_x g_y - \xi I_y h_x + \xi I_x h_y + \eta I(g_y \beta_x - g_x \beta_y) - \xi I(h_y \beta_x - h_x \beta_y)}{g_x h_y - g_y h_x} + I_t = 0 \tag{16}$$

Now let

$$p = \frac{g_x}{g_x h_y - g_y h_x} \qquad q = \frac{g_y}{g_x h_y - g_y h_x} \qquad a = \frac{g_y \beta_x - g_x \beta_y}{g_x h_y - g_y h_x} \tag{17}$$

$$r = \frac{h_x}{g_x h_y - g_y h_x} \qquad s = \frac{h_y}{g_x h_y - g_y h_x} \qquad b = \frac{h_y \beta_x - h_x \beta_y}{g_x h_y - g_y h_x} \tag{18}$$

we get

$$\eta I_y p - \eta I_x q - \xi I_y r + \xi I_x s + \eta I a - \xi I b + I_t = 0 \tag{19}$$

### 3.1 The Calibrated Case

In the case where the velocity of the background plane $\mathbf{c}(\mathbf{u}, t)$ is known, Equation (19) is linear in $p, q, r, s, a, b$. Given 7 or more successive frames of a video or 14 or more pairs of successive frames and the associated motion of the background plane, we can solve the equation point-wise over the entire image. Given

$n+1$ successive frames, we get $n$ equations in 6 variables at each point in the foreground plane, giving us an over-constrained linear system of the form

$$\mathbf{A}\mathbf{p} = \mathbf{m} \tag{20}$$

here the $i^{th}$ row of the matrix $\mathbf{A}$ is given by

$$\begin{aligned} \mathbf{A_i} &= \begin{bmatrix} \eta(i)I_y(i) & -\eta(i)I_x(i) & -\xi(i)I_y(i) & \xi(i)I_x(i) & \eta(i)I(i) & -\xi(i)I(i) \end{bmatrix} \\ \mathbf{m} &= [-I_t(1), \ldots, -I_t(n)]^\top \qquad \mathbf{p} = \begin{bmatrix} p & q & r & s & a & b \end{bmatrix}^\top \end{aligned} \tag{21}$$

The above system of equations is solved simply by the method of linear least squares. The corresponding values of $g_x, g_y, h_x, h_x, \beta_x, \beta_y$ can then be obtained as follows.

$$g_x = \frac{p}{ps - qr} \qquad g_y = \frac{q}{ps - qr} \qquad \beta_x = \frac{bp - ar}{ps - qr} \tag{22}$$

$$h_x = \frac{r}{ps - qr} \qquad h_y = \frac{s}{ps - qr} \qquad \beta_y = \frac{bq - as}{ps - qr} \tag{23}$$

### 3.2 The Uncalibrated Case

If the motion of the background image is not known, Equation (19) is bilinear in the variables $p, q, r, s, a, b$ and $\xi(t), \eta(t)$. If we consider frames $i = 1, \ldots, n+1$ and pixels $j = 1, \ldots, m$ in the foreground image, then we can rewrite Equation (19) in the following form

$$\mathbf{c}_i^\top \mathbf{A}_{ij} \mathbf{p}_j = 1 \qquad i = 1, \ldots, n \quad j = 1, \ldots, m \tag{24}$$

where

$$\begin{aligned} \mathbf{c}_i &= \begin{bmatrix} \xi(i) & \eta(i) \end{bmatrix}^\top \qquad \mathbf{p}_j = \begin{bmatrix} p(j) & q(j) & r(j) & s(j) & a(j) & b(j) \end{bmatrix}^\top \\ \mathbf{A}_{ij} &= \frac{-1}{I_t(i,j)} \begin{bmatrix} 0 & 0 & -I_y(i,j) & I_x(i,j) & 0 & -I(i,j) \\ I_y(i,j) & -I_x(i,j) & 0 & 0 & I(i,j) & 0 \end{bmatrix} \end{aligned}$$

This gives us $nm$ equations in $2n+6m$ variables, and we can solve them whenever $nm > 2n + 6m$.

Equation (24) is a system of bilinear equations, i.e. given $\mathbf{c}_i$, the system reduces to a linear system in $\mathbf{p}_j$ and vice versa. The overall problem however is highly non-linear. Using this observation a simple alternating procedure for solving the system can be used, which starting with a random initialization, alternates between updating $\mathbf{c}_i$ and $\mathbf{p}_j$ using linear least squares. Even with the fast iterative procedure, solving for the structure in this manner is not feasible. Hence we use a small slice through image stack and use it to solve for $\mathbf{c}_j$, which is then used as input to the calibrated algorithm described in the previous section to recover the flow over the entire foreground plane.

**Properties of the solution.** Observe that $\mathbf{Q}\mathbf{A}_{ij}\mathbf{R} = \mathbf{A}_{ij} \quad \forall i, j$ where

$$\mathbf{Q} = \begin{bmatrix} q_{12} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \quad \mathbf{R} = \frac{1}{q_{11}q_{22} - q_{12}q_{21}} \begin{bmatrix} q_{11} & 0 & q_{21} & 0 & 0 & 0 \\ 0 & q_{11} & 0 & q_{21} & 0 & 0 \\ q_{12} & 0 & q_{22} & 0 & 0 & 0 \\ 0 & q_{12} & 0 & q_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & q_{11} & q_{21} \\ 0 & 0 & 0 & 0 & q_{12} & q_{22} \end{bmatrix} \tag{25}$$

Hence, not knowing $\mathbf{c}(\mathbf{u}, t)$ gives rise to a solution which is ambiguous up to a $2 \times 2$ invertible transformation, and a corresponding ambiguity in the values of the various gradient estimates. Coupled with this is the translation ambiguity in $g$ and $h$. This gives rise to a six parameter family of ambiguities in the overall solution.

### 3.3 Integration

The methods described in the previous two sections give us estimates of the partial derivatives of $g(x, y)$, $h(x, y)$ and $\beta(x, y)$. The final step is integrating the respective gradient fields to recover the actual values of the functions. Reconstruction of a function from its first partial derivatives is a widely studied problem. We use an iterative least squares optimization that minimizes the reconstruction error [10].

## 4 Experiments

All experiments were done with video sequences of 200 frames each. The synthetic data were generated using a combination of MATLAB and POV-Ray, a public domain ray tracer. The real data was captured by placing a refracting object in front of an LCD screen, and imaging the setup using a firewire camera. Figure 2 illustrates the data acquisition setup. Calculation of image derivatives is

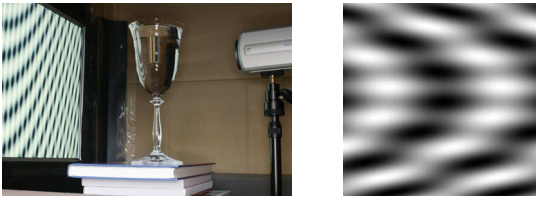


**Fig. 2.** (a) Shown above is a photograph of the data acquisition system used in our experiments. It consists of Samsung 19" LCD display screen on the left, a Sony DFW-VL500 firewire camera on the right and a refracting object between the screen and the camera. (b) shows a frame from the background image sequence for all our experiments.

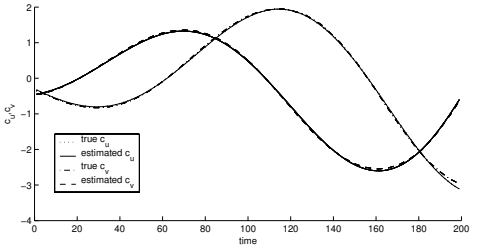



**Fig. 3.** Estimation of background motion in the uncalibrated case. This figure plots the true and estimated motion in the background plane. The two curves show motion along the $\xi(t)$ and $\eta(t)$ along the $u$ and $v$ axis respectively. As can be seen there is virtually no difference between the true and estimated values for $\xi(t)$ and $\eta(t)$.

very sensitive to noise. Smoothing was performed on the images using anisotropic diffusion [11], which has superior behavior along sharp edges as compared to Gaussian filtering. This is important for objects that cause light rays to be inverted, which in turn causes the optical flow across the boundary to be opposite sign; a naive Gaussian based smoothing procedure will result in significant loss of signal. The least squares estimation step in the calibrated estimation algorithm was made robust by only considering equations for which the temporal gradient term $I_t$ was within 85% of the maximum temporal gradient at that pixel over time. This choice results in only those constraints being active where some optical flow can be observed.

The boundary of refracting objects typically have little or no optical flow visible. This results in the refractive optical flow constraint breaking down along the boundary as well as certain medium interfaces. We mask these pixels out by considering the average absolute temporal gradient at each point in the foreground image plane and ignoring those pixels that fall below a chosen threshold. This results in a decomposition of the image into a number of connected components. All subsequent calculations are carried out separately for each connected component.

### 4.1 Results

We begin by considering a synthetic example to illustrate the performance of our algorithm in the calibrated and the uncalibrated case. The warping function used was $T(x, y) = (xe^{-(x^2+y^2)}, ye^{-(x^2+y^2)})^\top$ and the attenuation factor was $\alpha(x, y) = e^{-(x^2+y^2)}$. Figure 3 shows a comparison between the estimated and the true motion in the uncalibrated case. Figure 4 illustrates the results of the experiment. The estimated warp and attenuation functions are virtually identical to the true warp and attenuation functions. In the uncalibrated case, the estimated motion was disambiguated by doing a least squares fit to the ground truth for the purposes of illustration.

Figure 5 illustrates the result of applying the refractive structure of a glass with and without water inside it to the task of environment matting. Note that the region along the rim of the glass is missing. This is more so in case of the glass filled with water. These are the regions where the refractive optical flow equation breaks down. The black band in the case of the filled glass is due to a combination of the breakdown of the refractive optical flow equation along the air/water/glass interface and the finite vertical extent of the image. More results can be seen at `http://vision.ucsd.edu/papers/rof/`.

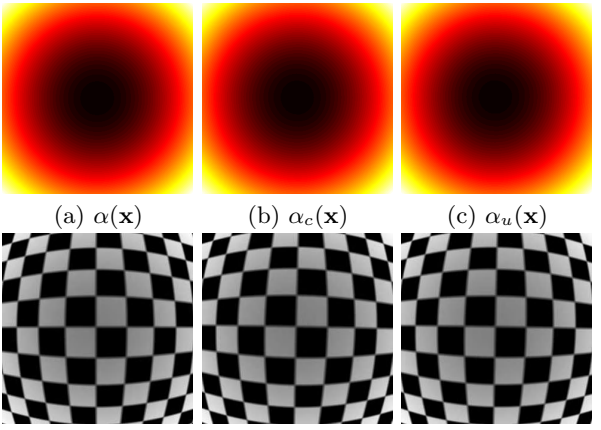



**Fig. 4.** This figure shows compares the performance of refractive structure estimation in the calibrated and the uncalibrated case. (a) and (d) show the true warp $T(\mathbf{x})$ applied to a checkerboard pattern and the attenuation factor $\alpha(\mathbf{x})$. (b) and (e) show the estimated warp and the alpha for the calibrated case, and (c) and (f) show the estimated warp and alpha for the uncalibrated case.

## 5 Discussion

We have introduced a generalization of the optical flow constraint, described methods for solving for the refractive structure of objects in the scene, and shown that this can be readily computed from images. We now comment on the limitations of our work and directions of future work.

First, our method does not address Fresnel and total internal reflection. This places a limit on our analysis to the case where all the illumination comes from behind the object being observed. Methods for recovering surface geometry from specular reflections are an active area of research and are better suited for this task [12,13,14].

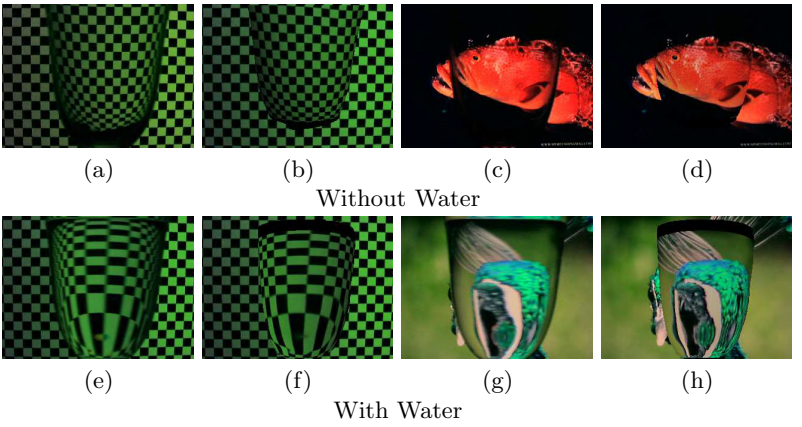



**Fig. 5.** Results of using the refractive structure for environment matting. (a), (c) show the true warping of a background image when an empty glass is placed in front of it, (b) and (d) show the estimated refractive structure applied to the same images. (e) and (g) show the true warping of a background image when a glass filled with water is placed in front of it, (f) and (h) show the estimated refractive structure applied to the same images.

Second, the presented approach, like [5,7], formulates the objective as determining a plane-to-plane mapping, i.e., it only informs us about how the object distorts a plane in 3-space. A more satisfactory solution will be to recover how the object distorts arbitrary light rays. We believe this is solvable using two or more views of the background plane and is the subject of our ongoing work.

The choice of the optical kernel that resulted in the single ray model was made for reasons of tractability. This however leaves the possibility of other optical kernels that account for multiple ray models.

Our experiments were carried out using a 19 inch LCD screen as the background plane. For most refracting objects, as their surface curvature increases, the area of the background plane that they project onto a small region in the foreground image can increase rapidly. If one continues to use a flat background one would ideally require an infinite background plane to be able to capture all the optical flow. Obviously that is not practical. An alternate approach is to use a curved surface as the background, perhaps a mirror which reflects an image projected onto it.

The current work only considers gray scale images; extensions to color or multi-spectral images is straightforward. There are two cases here: if the distortion $T$ is assumed to be the same across spectral bands, the generalization can be obtained by modeling $\alpha(\mathbf{x})$ as a vector valued function that accounts for attenuation in each spectral band. In case $T$ is dependent on the wavelength of light, each spectral band results is an independent version of Equation (12) and can be solved using the methods described.

Finally, the refractive optical flow equation is a very general equation describing optical flow through a distortion function. This allows us to address distortion not due to transmission through transparent objects, but also due to reflection from non-planar mirrored and specular surfaces. We believe that the problem of specular surface geometry can be addressed using this formalism, and this is also a subject of our future work.

**Acknowledgements.** The authors would like to thank Josh Wills and Kristin Branson for helpful discussions. This work was partially supported under the auspices of the U.S. DOE by the LLNL under contract No. W-7405-ENG-48 to S.B. and National Science Foundation grant IIS-0308185 to D.K.

## References

1. Irani, M., Rousso, B., Peleg, S.: Computing occluding and transparent motion. International Journal of Computer Vision **12** (1994) 5–16
2. Ju, S.X., Black, M.J., Jepson, A.D.: Skin and bones: Multilayer, locally affine, optical flow and regularization with transparency. In: CVPR 1996. (1996) 307–314
3. Levin, A., Zomet, A., Weiss, Y.: Learning to perceive transparency from the statistics of natural scenes. In: Neural Information Processing Systems, 2002. (2002)
4. Schechner, Y., Kiryati, N., Shamir, J.: Blind recovery of transparent and semireflected scenes. In: CVPR 2000. Volume 1., IEEE Computer Society (2000) 38–43
5. Chuang, Y.Y., Zongker, D.E., Hindorff, J., Curless, B., Salesin, D.H., Szeliski, R.: Environment matting extensions: Towards higher accuracy and real-time capture. In: Proc. of ACM SIGGRAPH 2000, ACM Press (2000) 121–130
6. Wexler, Y., Fitzgibbon, A.W., Zisserman, A.: Image-based environment matting. In: Proc. of the 13th Eurographics workshop on Rendering. (2002)
7. Zongker, D.E., Werner, D.M., Curless, B., Salesin, D.H.: Environment matting and compositing. In: Proc. of ACM SIGGRAPH 99, ACM Press (1999) 205–214
8. Murase, H.: Surface shape reconstruction of a nonrigid transparent object using refraction and motion. IEEE. Trans. on Pattern Analysis and Machine Intelligence **14** (1992) 1045–1052
9. Horn, B.K.P., Schunck, B.G.: Determining optical flow. Artificial Intelligence **17** (1981)
10. Horn, B.: Robot Vision. MIT Press (1986)
11. Perona, P., Malik, J., Shiota, T.: Anisotropic Diffusion. In: Geometry-Driven Diffusion in Computer Vision. Kluwer, Amsterdam (1995) 73–92
12. Savarese, S., Perona, P.: Local analysis for 3D reconstruction of specular surfaces. In: CVPR 2001. (2001) 738–745
13. Savarese, S., Perona, P.: Local analysis for 3D reconstruction of specular surfaces: Part II. In: ECCV 2002. (2002)
14. Oren, M., Nayar, S.: A theory of specular surface geometry. IJCV **24** (1997) 105–124

# Matching Tensors for Automatic Correspondence and Registration

Ajmal S. Mian, Mohammed Bennamoun, and Robyn Owens

School of Computer Science and Software Engineering
The University of Western Australia, Crawley, WA 6009, Australia,
{ajmal, bennamou, robyn}@csse.uwa.edu.au,
http://www.cs.uwa.edu.au

**Abstract.** Complete 3-D modeling of a free-form object requires acquisition from multiple view-points. These views are then required to be registered in a common coordinate system by establishing correspondence between them in their regions of overlap. In this paper, we present an automatic correspondence technique for pair-wise registration of different views of a free-form object. The technique is based upon a novel robust representation scheme reported in this paper. Our representation scheme defines local 3-D grids over the object's surface and represents the surface inside each grid by a fourth order tensor. Multiple tensors are built for the views which are then matched, using a correlation and verification technique to establish correspondence between a model and a scene tensor. This correspondence is then used to derive a rigid transformation that aligns the two views. The transformation is verified and refined using a variant of ICP. Our correspondence technique is fully automatic and does not assume any knowledge of the viewpoints or regions of overlap of the data sets. Our results show that our technique is accurate, robust, efficient and independent of the resolution of the views.

## 1 Introduction

Three dimensional modeling of objects has become a requirement in a large number of fields ranging from the entertainment industry to medical science. Various methods are available for scanning views of 3-D objects to obtain 2.5-D images in the form of a cloud of points (see Fig. 1), but none of these methods can completely model a free-form object with a single view due to self occlusion. Multiple overlapping views of the object must be acquired to complete the 3-D model. These views are then required to be registered in a common coordinate system, but before they can be registered, correspondence must be established between the views in their regions of overlap. Points on two different views that correspond to the same point on the object are said to be corresponding points. These correspondences are then used to derive an optimal transformation that aligns the views. The automatic correspondence problem is difficult to tackle due to two main reasons. First, there is no knowledge of the viewing angles and second, there is no knowledge about the regions of overlap of the views. The latter implies that every point on one view does not necessarily have a

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 495–505, 2004.
© Springer-Verlag Berlin Heidelberg 2004

corresponding point in the other view and that there is no a priori knowledge of correspondences.

Existing techniques of correspondence are based on various assumptions and are not fully automatic [1]. The classic Iterated Closest Point (ICP) algorithm [2], Chen and Medioni's algorithm [3] and registration based on maximizing mutual information [4] all require initial estimates. In case the initial estimate is not accurate, these techniques may not converge to the correct solution. Some techniques like the RANSAC-based DARCES [5] are based upon exhaustive search and are not efficient. Bitangent curve matching [6] calculates first order derivatives which are sensitive to noise and require the underlying surface to be smooth. Moreover, bitangent curves are global features and may not be fully contained inside the overlapping region of the views. Three tuple matching [7] calculates the first and second order derivatives which are sensitive to noise and require the underlying surfaces to be smooth. SAI matching [8] requires the underlying surfaces to be free of topological holes. Geometric histogram matching [9] makes use of a 3-D Hough transform [10] which is computationally expensive. Roth's technique [11] relies upon the presence of a significant amount of texture on the surface of the object for consistent extraction of feature points from their intensity images. Matching oriented points [12] uses spin image representation which is not unique and gives a lot of ambiguous correspondences. These correspondences must be processed through a number of filtration stages to prune out incorrect correspondences making the technique inefficient.

In this paper, we present a fully automatic correspondence technique which does not assume any prior knowledge of the view-points or the regions of overlap of the different views of an object. It is applicable to free-form objects and does not make assumptions about the shape of the underlying surface. Our technique is inspired by the spin image representation [13]. However, instead of making 2-D histograms of vertex positions, we represent the surface of the object in local 3-D grids. This results in a unique representation that facilitates accurate correspondences. The strength of our technique lies in the new representation scheme that we have developed. Our correspondence technique starts by converting two views of an object, acquired through a 3-D data acquisition system, into triangular meshes. Normals are then calculated for each point and triangular facet. Sets of two points along with their normals on each triangular mesh are then selected to define 3-D grids over the surface. The surface area and normal information in all the bins of each grid is then stored in a tensor. These tensors are matched to establish correspondences between the two views. Tensors that give the best match are then used to compute a rigid transformation that aligns the two views. This transformation is refined using a variant of the ICP algorithm [15].

The rest of this paper is organized as follows. In Section 2 we describe our new 3-D free-form object representation scheme. In Section 3 we explain the matching process to establish correct correspondences between the two views. Section 4 gives details of our experimental results. In Section 5 we discuss and analyze our results. Finally conclusions are given in Section 6.

## 2 A New Representation Scheme Based on Tensors

In this section we will describe our new tensor based 3-D free-form object representation scheme. Before we construct the tensors, the $n$ data points are first converted into triangular meshes and normals are calculated at each vertex and triangular facet. This information is stored in a data structure along with the neighbourhood polygons information for each point and each polygon. Next a set of two points, along with their normals are selected to define a 3-D coordinate basis. To avoid the $C_2^n$ combinatorial explosion of the points, we select points that are at a certain fixed distance from each other. This distance is defined as a multiple of the mesh resolution. In our experiments we have set this distance to four times the mesh resolution, which is far enough to make the calculation of the coordinate basis less sensitive to noise and close enough for both points to lie inside the region of overlap. To speed up the search for such points we consider points that are four edges away from each other. This can be easily performed by checking the fourth and fifth neighbourhood of the point under consideration. The center of the line joining the two points defines the origin of the new 3-D basis. The average of the two vectors defines the z-axis, since we want the z axis to be pointing away from the surface. The cross product of the two vectors defines the x-axis and finally the cross product of the z-axis with the x-axis defines the y-axis.

This 3-D basis and its origin is used to define a 3-D grid centered at the origin. Two parameters need to be selected, namely, the number of bins in the 3-D grid and the size of each bin. Varying the number of bins from less to more varies the representation from being local to global. We have selected the number of bins to be 10 in all the directions making the grid take the shape of a cube. The bin size defines the level of granularity at which the information about the object's surface is stored. The bin size is kept as a multiple of the mesh resolution because the mesh resolution is generally related to the size of the features on the object. In our experiments we have selected a bin size equal to the mesh resolution.

Once the 3-D grid is defined (see Fig. 1 and 2) the area of the triangular mesh intersecting each bin is calculated along with the average surface normal of the surface at that position. Next the angle between this surface normal and the z-axis of the grid is calculated. This angle is an estimate of the curvature of the surface at that point. This area and angle information is stored in a fourth order tensor which corresponds to a local representation of the surface in the 3-D cubic grid. To find the area of intersection of the surface with each cubic bin, we start from one of the two points that were used to define the 3-D grid and visit each triangle in its immediate neighbourhood. Since the points are approximately two mesh resolutions away from the origin they are bound to be inside the 3-D grid. The area of intersection of the triangle and a cubic bin of the grid is calculated using Sutherland Hodgman's algorithm [14]. Once all the triangles in the immediate neighbourhood of the point have been visited and their intersection with the grid bins has been calculated, the neighbourhood triangles of these triangles are visited. This process continues until a stage is reached when all the neighbouring triangles are outside the 3-D grid at which point the

computation is stopped. While calculating the area of intersection of a triangle with a cubic bin the angle between its normal and the z-axis is also calculated and stored in the fourth order tensor. Since more than one triangle can intersect a bin, the calculated area of intersection is added to the area already present in that bin, as a result of its intersection with another triangle. The angles of the triangular facets, crossing a particular bin, with the z-axis are averaged by weighting them by their corresponding intersection area with that bin.

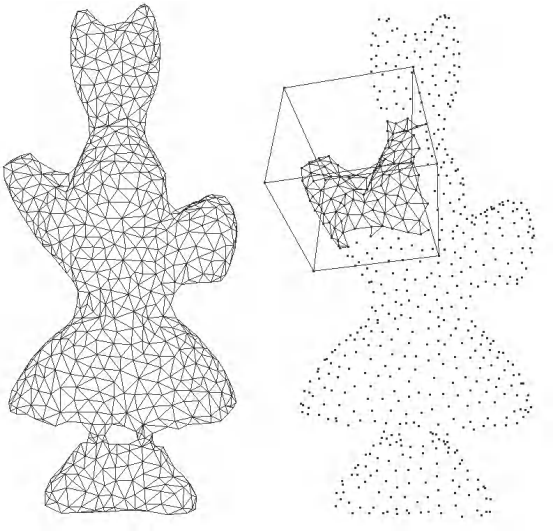


**Fig. 1.** Left: Data points of a view of the bunny converted into a triangular mesh. (Data courtesy of the Robotics Institute, CMU) Right: The cube represents the boundary of a 3-D grid. Only the triangles that contribute toward the tensor corresponding to this grid, as shown, are considered in the calculation of the tensor corresponding to this grid.

## 3 Matching Tensors for Correspondence and Registration

Since a tensor corresponds to a representation of the 3-D surface inside an object centered grid, different views of the same surface will have similar tensors. Minor differences may exist between these tensors as a result of different possible triangulations of the same surface due to noise and variations in sampling. However, corresponding tensors will have a better match as compared to the non-corresponding tensors. We use the linear correlation coefficient to match tensors. Corresponding tensors will give a high correlation coefficient and can easily be differentiated.

To establish correspondence between a model view and a scene view of an object, first the tensors for all the point pairs of the model (that are four times

the mesh resolution apart) are calculated. Restricting the selection of point pairs significantly reduces the number of possible pairs from $C_2^n$. The search for such points is speeded up by searching the fourth and fifth neighbourhood of the points only. Next a point is selected at random from the scene and all possible points that can be paired with it are identified. A tensor is then calculated for the first point and one of its peers. This tensor is then matched with all the tensors of the model. If a significant match is found, the algorithm proceeds to the next stage of verification, else it drops this tensor and calculates another tensor using the first point and another one of its remaining peers. The tensors are matched only in those bins where both tensors have surface data (this approach has also been used by Johnson [12]). This is done to cater for situations where some part of the object may be occluded in one view. Matching proceeds as follows. First, the overlap ratio $R_O$ of the two tensors is calculated according to Equation 1. If $R_O$ is greater than a threshold $t_r$, the algorithm proceeds to calculate the correlation coefficient of the two tensors in their region of overlap. If $R_O$ is less than $t_r$, the model tensor is not considered for further matching. In our experiments we found that $t_r = 0.6$ gave good results.

$$R_O = \frac{\sum I_{sm}}{\sum U_{sm}} \tag{1}$$

In this Equation $I_{sm}$ is the intersection of the occupied bins of the scene and the model tensor. $U_{sm}$ is the union of the occupied bins of the scene and the model tensor.

If the correlation coefficient of the scene tensor with some model tensors is significantly higher than the correlation coefficients with the remaining model tensors, then all such model tensors are considered to be potential correspondences. Such model tensors are taken to be those having correlation coefficient two standard deviations higher than the mean correlation coefficient of the scene tensor with all the model tensors. The best matching tensor is verified first. Verification is performed by transforming one of the two scene points, used to calculate the tensor, to the model coordinate system. This transformation is calculated by transforming the corresponding 3-D basis of the scene tensor to the 3-D basis of the model tensor (Eqn. 2 and Eqn. 3).

$$\mathbf{R} = \mathbf{B_s^T B_m} \tag{2}$$

$$\mathbf{t} = \mathbf{O_m} - \mathbf{O_s R} \tag{3}$$

$\mathbf{B_m}$ and $\mathbf{B_s}$ are the matrices of coordinate basis used to define the model and scene tensors respectively. $\mathbf{O_m}$ and $\mathbf{O_s}$ are the coordinates of the origins of the model and scene grids in the coordinate basis of the entire scene and model respectively. $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector that will align the scene data with the model data.

Next the distance between the transformed scene point and its corresponding point in the model tensor is calculated. If this distance is below a threshold $d_{t1}$, i.e. the scene point is close to its corresponding model point, the verification process proceeds to the next step, else the model tensor is dropped and the next

model tensor with the highest correlation coefficient is tested. Figure 2 shows an incorrect transformation calculated as a result of matching tensors. The two points that were used to calculate the tensors are connected by a line. These points are not close to their counter-parts in the other tensor, representing a poor tensor match. Figure 2 also shows a correct transformation calculated as a result of matching tensors. Here the two points are very close to their counter-parts in the other tensor representing a good tensor match. In our experiments we set $d_{t1}$ equal to one fourth of the mesh resolution to ensure that only the best matching tensors pass this test. If all model tensors fail this test, another set of two points is selected from the scene and the whole process is repeated.

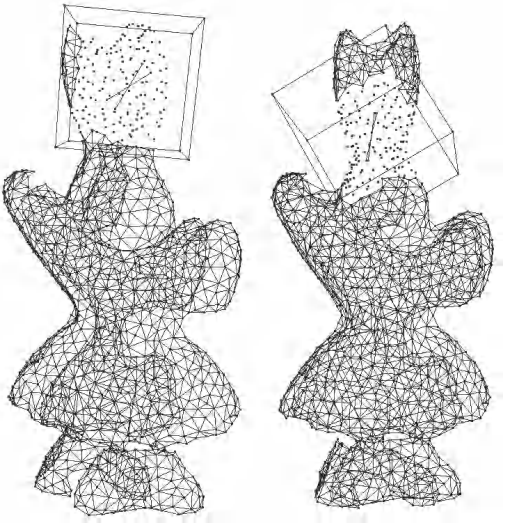


**Fig. 2.** Two views of the bunny registered using a single set of matching tensors. The bounding box is the region where the scene and model tensors are matched. The points used to calculate the 3-D basis are joined by a line. Left: These points are not close to their counter-parts resulting in an inaccurate transformation. Right: The points are close to their counter-parts in the other view, resulting in a good tensor match and an accurate transformation.

In case the distance between the two pairs of points is less than $d_{t1}$, all the scene points are transformed to the model coordinate system using the same transformation. The transformation resulting from a single set of good matching tensors is accurate enough to establish scene point to model point correspondences on the basis of nearest neighbour. The search for nearest neighbour starts from the scene points that are connected directly to one of the initial points used to define the 3-D basis. Scene points that have a model point within a distance of $d_{t2}$ are turned into correspondences. We chose $d_{t2}$ equal to the mesh resolution in our experiments. $d_{t2}$ is selected considerably higher than $d_{t1}$ since the initial transformation has been calculated based on a single set of matching tensors.

Even a small amount of error in this transformation will cause greater misalignment between the scene and the model points that are far from the origin of the tensors. Next correspondences are found for more scene points that are directly connected to the points for which correspondences have recently been found. This process continues until correspondences are spread throughout the mesh and no more correspondences can be found. If the total number of correspondences at the end is more than half the total number of scene points, the initial transformation given in Equations 2 and 3 is accepted and refined by applying another transformation calculated from the entire set of correspondences found during the verification process.

## 4 Experimental Results

We have performed our experiments on a large data set. The results of only four objects are reported in this paper. The data set (in the form of a cloud of points) of the first three of these objects namely, the bunny, the truck and the robot was provided by the Robotics Institute, Carnegie Mellon University, whereas the data of Donald Duck was acquired using the Faro Arm acquisition system in our laboratory. The gray scale pictures of these objects are shown in Figure 3. Three views of the first three objects were taken and our automatic correspondence algorithm was applied to register these views. Figure 4 shows the results of our experiments. Each row of Figure 4 contains a different object. The first three columns of Figure 4 contain the three different views of these objects and the fourth column contains all three views registered in a common coordinate frame. The registered views are shown in different shades so that they can be differentiated after alignment.

**Fig. 3.** Gray scale pictures of the bunny, the truck, the robot and Donald Duck used in our experiments.

We have also tested our algorithm on data sets where each view is acquired at different resolutions and in each case it resulted in an accurate registration. This shows that our algorithm is independent of the resolution of the data. Unlike spin image matching our correspondence algorithm does not require uniform mesh resolution and is therefore robust to variations in resolution within a single view. Figure 5 shows the result of our algorithm on the Donald Duck data set

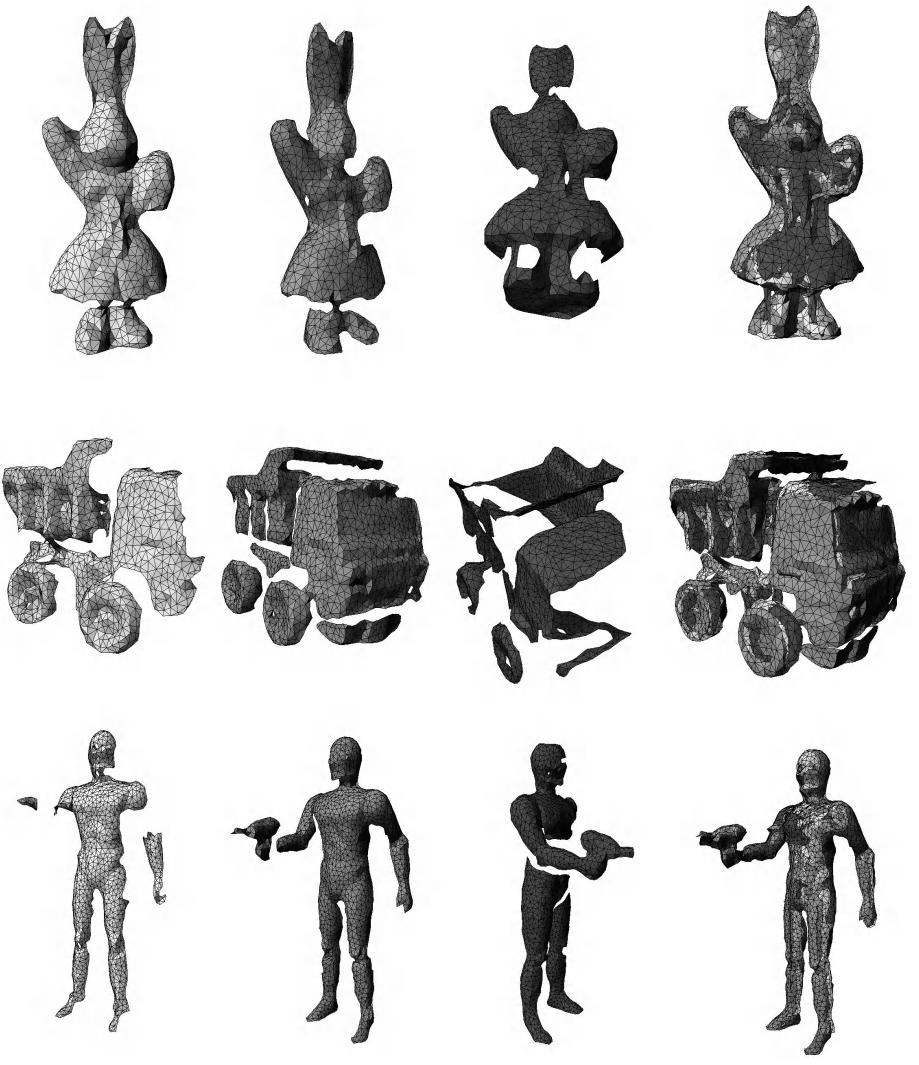


**Fig. 4.** Registration results. Each row contains a separate object, namely the bunny, the truck and the robot. The first three columns of the Figure represent three views of the respective objects in different grey shading. The last column shows the registered views. Notice the contribution of each view shown in different shadings in the registered view.

that has an extremely non-uniform mesh resolution with edge lengths varying from a minimum of $0.2mm$ to a maximum of $12.8mm$ and a standard deviation of $2.4mm$. The registration result in this case is an indication of the extent of robustness of our algorithm to variations in the resolution of data set. Such variations are commonly expected from all sensors when there are large variations

in the orientation of the surface. Data points of a surface patch that is oblique to the sensor will have low density as compared to a surface patch that is vertical to the sensor.

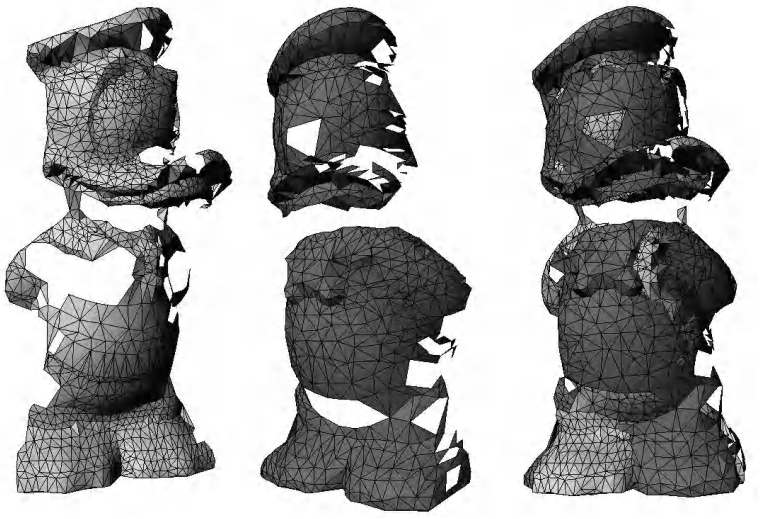


**Fig. 5.** Result of the algorithm applied to two views of Donald Duck having non-uniform mesh resolution and patches of missing data. Accurate registration is obtained in the presence of missing data. The missing data is due to the difficulty in scanning with a contact sensor (the Faro Arm in our case).

## 5 Discussion and Analysis

Our tensor based representation scheme is robust to noise due to the following reasons. The 3-D basis is defined from two points that are four mesh resolutions apart. The origin is defined by the center of the line joining the two points which reduces the effect of noise. The z-axis is defined as the average of the normals of these points, hence reducing the effect of noise and variations in surface sampling. Quantization is performed by dividing the surface area into the bins of a 3-D grid. This significantly reduces the effect of different possible triangulations of the surface data. The use of a statistical matching tool (in our case the correlation coefficient) performs better in the presence of noise as compared to linear matching techniques. All these factors ensure that the tensors representing the same surface in two different views will give high similarity as compared to tensors representing different surface regions.

We have taken the following measures to ensure that our algorithm is efficient in terms of memory consumption and performance. First, to avoid the combinatorial explosion of pairing points, we only consider points that are four mesh

resolutions apart with some tolerance. This restricts the possible pairs of points to $O(n)$ instead of $O(n^2)$. Next, to calculate the area of the mesh inside the individual bins of the 3-D grid, instead of visiting each bin of the grid, we start from one of the points used to define it and consider its neighbouring polygons that are inside the grid. These polygons are visited one at a time and their area of intersection with the bins is calculated using an efficient algorithm (Hodgman [14]).

During the matching phase, in order to speed up the process, two tensors are only matched if their overlap ratio $R_O$ is more than 0.6. If a match with a high correlation coefficient is found, it is verified by transforming only one of the scene points, used to define the 3-D basis. If the distance of the transformed point is less than $d_{t1}$ from its corresponding model point, the transformation is accepted, otherwise it is rejected. This verification step is very fast since the rotation matrix and the translation vector can easily be calculated from Equations 2 and 3. Choosing $d_{t1}$ equal to 1/4th of the mesh resolution ensures that only a good match passes this test. This verification step almost always identifies an incorrect tensor match and saves the algorithm from proceeding to the verification stage. Verification of the transformation involves the search for the nearest neighbour of every scene point, which is computationally expensive. Our algorithm does not find a list of correspondences and pass them through a series of filtration steps as in the case of the spin images approach [12]. Instead it selects a scene tensor and finds its matching tensor in the model. If the match passes the above explained verification step, it proceeds to verify the transformation derived from Equations 2 and 3. This transformation is refined and the algorithm stops. The algorithm does not have to visit every possible correspondence and is therefore less computationally expensive.

Once the views are registered they can easily be integrated and reconstructed to form a single smooth and seamless surface. We have intentionally presented our raw results after applying registration only so that the accuracy of our algorithm could be analyzed. In the future, we intend to use this algorithm for multi-view correspondence and registration. An extension of this work is to achieve multi-view correspondence without any prior knowledge of the ordering of the views.

## 6 Conclusion

We have presented a novel 3-D free-form object representation scheme based on tensors. We have also presented a fully automatic correspondence and registration algorithm, based on our novel representation. Our algorithm makes no assumption about the underlying surfaces and does not require initial estimates of registration or the viewing angles of the object. The strength of our algorithm lies in our robust representation scheme based on fourth order tensors. We have presented an effective and efficient procedure for matching these tensors to establish correct correspondence between a model and a scene surface. The algorithm has been tested on different data sets of varying mesh resolution and our results show the effectiveness of the algorithm.

**Acknowledgments.** We are grateful to the Robotics Institute, Carnegie Mellon University, USA for providing us with the data used in our experiments. This research is supported by ARC grant number DP0344338.

## References

1. Mian, A. S., Bennamoun, M., Owens, R.: Automatic Correspondence for 3D Modeling: An Extensive Review. Submitted to a journal, (2004)
2. Besl, P.J., McKay, N.D.: Reconstruction of Real-world Objects via Simultaneous Registration and Robust Combination of Multiple Range Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2 (1992) 239–256
3. Chen, Y., Medioni, G.: Object Modeling by Registration of Multiple Range Images. IEEE International Conference on Robotics and Automation (1991) 2724–2729
4. Rangarajan, A., Chui, H., Duncan, J.S.: Rigid point feature registration using mutual information. Medical Image Analysis, Vol. 3, No. 4, (1999) 425–440
5. Chen, C., Hung, Y., Cheng, J. RANSAC-Based DARCES: A New Approach to Fast Automatic Registration of Partially Overlapping Range Images. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 11 (1991) 1229–1234
6. Wyngaerd, J., Gool, L., Koth, R., Proesmans, M.: Invariant-based Registration of Surface Patches. IEEE International Conference on Computer Vision, Vol. 1 (1999) 301–306
7. Chua, C.S., Jarvis R.: 3D Free-Form Surface Registration and Object Recognition. International Journal of Computer Vision, Vol. 17, (1996) 77–99
8. Higuchi K., Hebert M., Ikeuchi K.: Building 3-D Models from Unregistered Range Images. IEEE International Conference on Robotics and Automation, Vol. 3, (1994) 2248–2253
9. Ashbrook, A.P., Fisher, R.B., Robertson, C., Werghi, N.: Finding Surface Correspondence for Object Recognition and Registration Using Pairwise Geometric Histograms. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 2 (1998) 674–686
10. Stephens, R.S.: A probabilistic approach to the Hough transform. British Machine Vision Conference (1990) 55–59
11. Roth, G.: Registering Two Overlapping Range Images. IEEE International Conference on 3-D Digital Imaging and Modeling (1999) 191–200
12. Johnson, A.E., Hebert, M.: Surface Registration by Matching Oriented Points. International Conference on Recent Advances in 3-D Imaging and Modeling (1997) 121–128
13. Johnson, A.E.: Spin Images: A Representation for 3-D Surface Matching. PhD. Thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213 (1997)
14. Foley, J., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics-Principles and Practice. Addison-Wesley, Second Edition (1990)
15. Zhang, Z.: Iterative Point Matching for Registration of Free-form Curves and Surfaces. International Journal of Computer Vision, Vol. 13, No. 2, (1994) 119–152

# A Biologically Motivated and Computationally Tractable Model of Low and Mid-Level Vision Tasks

Iasonas Kokkinos[1], Rachid Deriche[2], Petros Maragos[1], and Olivier Faugeras[2]

[1] School of Electrical and Computer Engineering,
National Technical University of Athens, Greece
{jkokkin,maragos}@cs.ntua.gr
[2] INRIA, 2004 route des Lucioles, BP 93, 06902 Sophia-Antipolis Cedex, France
{Rachid.Deriche,Olivier.Faugeras}@sophia.inria.fr

**Abstract.** This paper presents a biologically motivated model for low and mid-level vision tasks and its interpretation in computer vision terms. Initially we briefly present the biologically plausible model of image segmentation developed by Stephen Grossberg and his collaborators during the last two decades, that has served as the backbone of many researchers' work. Subsequently we describe a novel version of this model with a simpler architecture but superior performance to the original system using nonlinear recurrent neural dynamics. This model integrates multi-scale contour, surface and saliency information in an efficient way, and results in smooth surfaces and thin edge maps, without posterior edge thinning or some sophisticated thresholding process. When applied to both synthetic and true images it gives satisfactory results, favorably comparable to those of classical computer vision algorithms. Analogies between the functions performed by this system and commonly used techniques for low- and mid-level computer vision tasks are presented. Further, by interpreting the network as minimizing a cost functional, links with the variational approach to computer vision are established.

## 1 Introduction

A considerable part of the research in the computer vision community has been devoted to the problems of low and mid level vision; even though these may seem to be a trivial task for humans, they are intrinsically difficult and the human visual system still outperforms by far the state-of-the-art. Therefore, its mechanisms could serve as a pool of ideas for computer vision research.

In this paper we propose a biologically motivated system for edge detection and image smoothing, apply it to real and synthetic systems, and explain it in computer vision terms. Our starting point is the system developed by Stephen Grossberg and his collaborators through a series of papers, which can be seen as the backbone for many researchers' models, e.g.[19,22]. While keeping the philosophy of the model intact, we propose using less processing stages and recurrent neural dynamics, resulting in a simpler yet more efficient model. Further, by

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 506–517, 2004.
© Springer-Verlag Berlin Heidelberg 2004

building upon previous work on the connections between neural networks and variational techniques, we interpret the network model in variational terms.

Is section 2 we shall briefly present the architecture proposed by S. Grossberg and his collaborators for mid-level vision tasks, while in section 3 we shall propose our model, that uses a simpler and more efficient version of this architecture. In section 4 the interpretation of our model in computer vision terms is presented.

## 2 A Review of the Boundary/Feature Contour Systems

The model proposed by Stephen Grossberg and his collaborators in a series of papers [10,11,12,13,14], known as the FACADE (Form And Colour and DEpth) theory of vision is a versatile, biologically plausible model accounting for almost the whole of low- and mid-level vision, starting from edge detection and ending at segmentation and binocular vision (see references in [31]). Most of the ideas are intuitively appealing and relatively straightforward; however, as a whole, the system becomes complicated, in terms of both analysis and functionality. This is natural, though, for any model of something as complicated as our visual system; the model's ability to explain a plethora of psychophysical phenomena [11,31] in a unified way offers support for its plausibility and motivation for studying it in depth, trying to relate and compare it with computer vision techniques.

We have implemented the two essential components of the FACADE model, namely the Boundary Contour System (B.C.S.) and the Feature Contour System (F.C.S.). The B.C.S. detects and amplifies the coherent contours in the image, and sends their locations to the F.C.S.; subsequently, the F.C.S. diffuses its image-derived input apart from the areas where there is input from the B.C.S., signaling the existence of an edge; see Fig. 1(a) for a 'road map' of the system.

An Ordinary Differential Equation (ODE) commonly used by S. Grossberg to determine the inner state $V$ of a neuron as a function of its excitatory ($E$) and inhibitory ($I$) input is:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = -AV + (C - V)E - (D + V)I, \quad E = \sum_{n=1}^{N} w_n^e U_n,\ I = \sum_{n=1}^{N} w_n^i U_n \qquad (1)$$

where $A$ is a passive decay term modeling the leakage conductance of a neuron; $C$ and $-D$ are the maximal and minimal attainable values of $V$ respectively; $U_n$ are the outputs of neurons $1, \ldots, N$; $w_n^{\{e,i\}}$ are the excitatory/inhibitory synaptic weights between neuron $n$ and the current neuron. The inner state $V$ of the neuron is related to its output -a mean firing rate- $U$ by a sigmoidal function; a reasonable choice is $U = G(V) = 1/(1 + \exp(-\beta(V - 1/2))$ so that for $V = 0$ we have a low output $U$. Equation (1) is closer to the neural mechanisms of excitation and inhibition than the common 'weighted sum of inputs' model and can account for divisive normalization and contrast invariance [3], which are helpful in visual processing tasks.

**The Boundary Contour System**

The outputs of cells in the B.C.S. are calculated as the rectified steady-state outputs of (1), so that all one needs to define are the excitatory and inhibitory

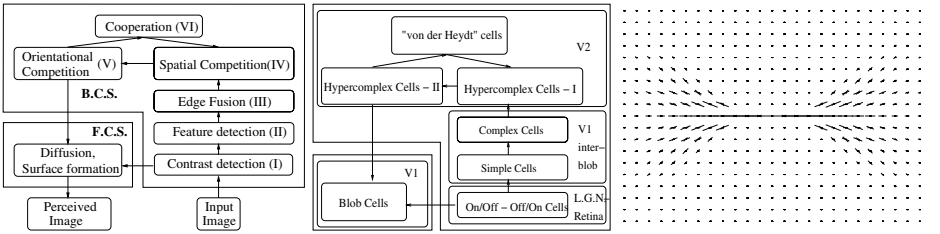



**Fig. 1.** (a) A block diagram of the B.C.S./F.C.S. architecture. (b) corresponding areas in the visual system (c) The shape of the lobes used for saliency detection in the horizontal direction, at stage VI. The 'needle' lengths are proportional to the weight assigned to each edge element in the corresponding direction and location

inputs; the processing stages used in the B.C.S. can be summarized as follows:

**Stage I** *Contrast Detection*: This stage models retinal cells, which exist in two varieties: one that responds to a bright spot surrounded by dark background, On/Off cells, and another that responds to the a dark spot on a bright background, Off/On cells. The excitatory input to an On/Off cell is modeled by convolving the input image $F$ with a Gaussian filter of spread $\sigma_1$ and the inhibitory input is modeled by convolving with a Gaussian filter of spread $\sigma_2 < \sigma_1$; the roles of excitatory and inhibitory terms are swapped for Off/On cells.

**Stage II** *Elementary Feature Detection*: This is the function accomplished by simple cells in cortex area V1; their function is modeled by assuming each cell's excitatory/inhibitory input equals the convolution of the previous stage's outputs with a spatially offset and elongated Gaussian, with principal axis along their preferred orientation. Specific values used for the centers and spreads of this stage can be found in [31,14]. Given that there can be only positive cells responses, two varieties of cells are needed at this stage, one responding to increase and another to decrease in activation along their direction.

**Stage III** *Cue Integration*: At this stage, the outputs of simple cells responding to increases and decreases in image intensity are added to model complex cells, which are known to respond equally well to both changes. If color and/or depth information is used as well, this is where edge fusion should take place, based on biological evidence.

**Stage IV** *Spatial Competition*: The outputs of the previous stage are 'thinned' separately for each feature map, using Gaussian filters of different spreads to excite and inhibit each cell. The convolution of the complex cells' outputs with a Gaussian of large spread (resp. small spread) is used as an inhibitory (resp. excitatory) input. A novel term that comes into play is the feedback term from Stage VI, which drives the competition process towards globally salient contours; this results in a modified version of equation (1), where the constant feedback term is added to the competition process.

**Stage V** *Orientational Competition*: This time the competition takes place among neurons with the same position but different orientations; the goal is to derive an unambiguous edge map, where there can be only one tangent direction for a curve passing through a point. The excitatory input to a cell is the

activation of the Stage IV cell at the same location and orientation, while its inhibitory input comes from Stage IV cells with different orientations; the inhibitory weights depend on the angles between the orientations, becoming maximal for perpendicular orientations and minimal for almost parallel.

**Stage VI** *Edge Linking, Saliency Detection*: This is where illusory contours emerge; each neuron pools evidence from its neighborhood in favor of a curve passing through it with tangent parallel to its preferred orientation $\theta$, using connections shown schematically in Fig. 1(c); mathematical expressions can be found in [14,31]. The output of a neuron at this stage is positive only when both of its lobes are activated, which prevents the extension of a line beyond its actual end. The output is sent to stage IV, thus resulting in a recurrent procedure, which detects and enhances smooth contours.

**The Feature Contour System**

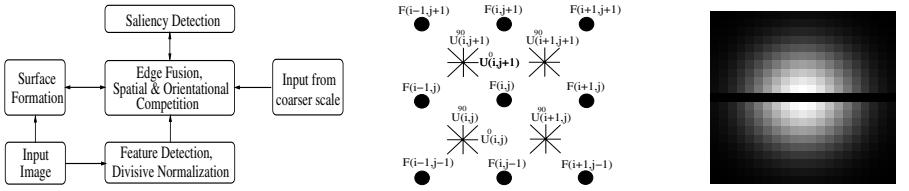
This is the complementary system to the B.C.S., where continuous brightness maps are formed, leading to the perception of surfaces. The image related input to this system is the output of the On/Off-Off/On cells of stage I; the formation of continuous percepts is accomplished by a process that diffuses the activities of the neurons, apart from locations where there is a B.C.S. signal, keeping the activities of neurons on the sides of an edge distinct. Specifically, the equations used to determine the activations of neurons $s(i,j)$ corresponding to On/Off or Off/On input signals, are:

$$\frac{\mathrm{d}}{\mathrm{d}t}s(i,j) = -As(i,j) + X(i,j) + \sum_{(p,q)\in N_{\{i,j\}}} (s(p,q) - s(i,j))P_{(p,q),(i,j)} \qquad (2)$$

where $X(i,j)$ is the steady-state output of Stage I (On/Off-Off/On cells) and $N_{\{i,j\}} = \{(i-1,j),(i+1,j),(i,j-1),(i,j+1)\}$. $P_{(p,q),(i,j)}$ is a decreasing function of the edge strength between pixels $(i,j),(p,q)$, which is computed using the steady state values of the B.C.S. stage V neurons. Physiological aspects of the mechanisms underlying the computation performed at this stage are discussed in [10], ch I, §24-26. Solving this system of ODEs with initial conditions the values of the On/Off- Off/On cells accomplishes the anisotropic diffusion of the On/Off cells activations, where the B.C.S. outputs determine where the diffusion becomes anisotropic; the perceived surfaces are modeled as the differences of the two steady-state solutions. According to the B.C.S./F.C.S. model, the perceived edges are at the points of intensity variation of the converged F.C.S. outputs.

## 3 A Simple and Efficient Biologically Plausible Model

In our implementations of the above stages we faced many problems: because so many parameters and interdependent stages are involved, the whole system becomes hard to tame; despite all our efforts, it was not possible to achieve a consistent behavior even for a limited variety of images. Since our first goal was to compare this system with some commonly used techniques for computer vision, and to see whether we can achieve an improved performance, we decided to simplify some of its stages, while keeping as close as possible to the original




(a) (b) (c)

**Fig. 2.** (a) The architecture of the new model (b) Relative positions of neurons and (c) Shape of lateral connections between neurons of the same orientation

architecture. Apart from the complexity and efficiency problems, most of our changes were inspired by the research experience accumulated among the computer vision community, which can be summarized as follows: (i) Edge thinning (stages IV-V) is an inherently *recurrent* process and not a feedforward one. The recurrent feedback term used in stage IV does not significantly help thin edges, even though it does help create new ones; recurrence should be used within each layer, using *lateral* connections among the neurons.(ii) The F.C.S. should be let to interact with the B.C.S., since the finally perceived edges are derived from the F.C.S.; therefore, it should play a role in the contour formation process, enhancing the more visible contours and exploiting region-based information. (iii) Results from coarse scales should be used to drive the edge detection process at finer scales. In the original B.C.S./F.C.S. architecture multiple scale results were simply added after convergence.

The model we propose, shown schematically in Fig. 2(a), is similar to the model of vision proposed by S. Grossberg, but is simpler and more efficient from a computational point of view. The architecture is similar to that proposed in [19], which, however, did not include a high level edge grouping process and a bottom up edge detection process, while the system was focused on texture segmentation. Our model performs both contour detection and image smoothing so it is different from other biologically plausible models like [22,20,16] that deal exclusively with boundary processing. We now describe in detail our model:

**Stage I'**: *Feature Extraction, Normalization.* The first two stages (contrast & feature detection) of the B.C.S. are merged into one, using the biologically plausible Gabor filterbank [6] described in [19]; this filterbank includes zero-mean odd-symmetric filters while the parameters of the filters are chosen to comply with measurements of simple cell receptive fields. If we ignore the normalization and rectification introduced by (1), the cascade of the first two stages of the B.C.S. can be shown to be similar to filtering with a Gabor filterbank [31]. To account for divisive normalization, we used the mechanism of shunting inhibition, as in [3], where the feedforward input to each cell is determined from the beginning using a convolution with a Gabor filter, while the shunting term is dynamically changing, based on the activations of neighboring cells. The terms contributing to the shunting inhibition of neuron $(i, j)$ are weighted with a Gaussian filter with spread equal to that of the Gabor filters; contributions from different orientations than $\theta$ are given equal weights, so that the equation driving the activity of a simple cell becomes:

$$\frac{\mathrm{d}V^{\theta,\sigma}_{\{o,e\}}}{\mathrm{d}t} = -AV^{\theta,\sigma}_{\{o,e\}} + (C - V^{\theta,\sigma}_{\{o,e\}})[F * \Psi^{\theta,\sigma}_{\{o,e\}}] - V^{\theta,\sigma}_{\{o,e\}} \sum_{\theta,o,e} G_\sigma * U^{\theta,\sigma}_{\{o,e\}} \quad (3)$$

where $\Psi^{\theta,\sigma}_{\{o,e\}}$ are odd/even symmetric filters, with orientation preference $\theta$ at scale $\sigma$ as in [19], while $U^{\theta,\sigma}_{\{o,e\}} = \max(V^{\theta,\sigma}_{\{o,e\}}, 0)$ are the time varying outputs of odd/even cells. For biological plausibility, we separate positive and negative simple cell responses, and perform the normalization process in parallel for each 'half', with the other half contributing to the shunting term; none of this appears in the above equations, for the sake of notational clarity.

**Stage II'**: *Edge Thinning, Contour Formation.* At the next stage, an orientational energy-like term [1,25] is used as feedforward input:

$$E_{\theta,\sigma} = \sqrt{(U^{\theta,\sigma}_o)^2 + (U^{\theta,\sigma}_e)^2}\ . \quad (4)$$

The difference lies in that simple cell outputs have already been normalized before, which results in a contrast invariant edge detector.

Edge sharpening in space and orientation is accomplished simultaneously with lateral connections among neurons of this stage. Specifically, the form of the lateral interactions among cells of the same orientation is modeled as the difference between an isotropic Gaussian and an elongated Gaussian whose longer axis is along the preferred orientation of the cells; this way fat edges are avoided, allowing a single neuron to be active in its vertical neighborhood, while collinear neurons support the formation of a contour. For example, for an horizontally oriented neuron located at $[i, j]$ the inhibitory connection strengths with neurons of the same orientation are (see also Fig.2(c)):

$$W^{0,0}_{[i,j]}[k,l] = \exp\left(-\left\{\frac{(i-k)^2 + (j-l)^2}{2\sigma_1^2}\right\}\right) - b\exp\left(-\left\{\frac{(i-k)^2}{2\sigma_1^2} + \frac{(j-l)^2}{2\sigma_2^2}\right\}\right) \quad (5)$$

where $\sigma_1 > \sigma_2$. We use $W^{\theta,\phi}_{[i,j]}[k,l]$ to express the strength with which the neuron at $[k,l]$ with orientation $\phi$ inhibits the one at $[i,j]$ with orientation $\theta$. The lateral inhibitory connections among complex cells of different orientational preferences are determined not only by the spatial but also by the orientational relationship between two neurons: almost collinear neurons inhibit each other, so that a single orientation dominates at each point, while perpendicular neurons do not interact, so that at corners the edge map does not break up. Such interactions among a neuron located at $[i, j]$ with horizontal orientation and a neuron located at $[k, l]$ with orientation $\theta$ can be expressed as:

$$W^{0,\theta}_{[i,j]}[k,l] = |\cos(\theta)| \exp\left(-\left\{\frac{(i-k)^2}{2\sigma_1^2} + \frac{(j-l)^2}{2\sigma_2^2}\right\}\right) \quad (6)$$

Given that this time the competition process aims at cleaning the edge map, and not thinning it, we do not choose a specific direction for the Gaussian, so we set $\sigma_1 = \sigma_2$. Choosing this particular type of connection weights was based on simplicity, convenience and performance considerations [31].

Apart from the coarsest scale, the steady state outputs of the immediately coarser scale, $U^{\theta,\sigma+1}$, are used as a constant term that favors the creation of edges

at specific locations. The feedback term, denoted by $T(t)$, that is calculated at the following stage, is used throughout the evolution process, facilitating the timely integration of high-level information. Otherwise, if the system is let to converge before using the feedback term, it may be driven to a local minimum, so that it may be hard to drive it out of it. In addition to that, we coupled the evolution of the curve process with the evolution of the surface process, $S$; the magnitude of the directional derivative $|\nabla S_{\theta^\perp}|$ of the surface process perpendicular to $\theta$ helps the formation of sharp edges at places where the brightness gradient is highest. If the output of a neuron with potential $V^{\theta,\sigma}$ is $U^{\theta,\sigma} = g(V^{\theta,\sigma})$ the evolution equation for the activation of neuron $[i,j]$ at this stage is written as:

$$\frac{\mathrm{d}V^{\theta,\sigma}}{\mathrm{d}t} = -AV^{\theta,\sigma} + (C - V^{\theta,\sigma})I - (V^{\theta,\sigma} + D)\sum_{\theta'}\sum_{k,l} W^{\theta,\theta'}_{[i,j]}[k,l]U^{\theta',\sigma}[k,l] \quad (7)$$

$$\text{where} \quad I(t) = [c_1 E + c_2 U^{\theta,\sigma+1} + c_3 T(t) + c_4 |\nabla S_{\theta^\perp}(t)|^2] \ .$$

The cues determining the excitatory input to the neuron, $I$, are bottom up - $E$, region based - $|\nabla S_{\theta^\perp}(t)|$, coarse scale -$U^{\theta,\sigma+1}$ and top-down -$T$. The weights, $c_1, c_2, c_3, c_4$, have been determined empirically [31] so as to strike a balance between the desire of being able to produce an edge in the absence of bottom-up input ($E$) and of keeping close to available bottom-up input.

**Stage III'**: *Salience Computation.* This layer's outputs are calculated as the product of the two lobes' responses, which are continuously updated, resulting in a process that is parallel and continuously cooperating with Stage-II. Even though it is not clear *how* multiplication can be performed in a single cell, there is strong evidence in favor of multiplication being used in mechanisms as gain modulation; see also [22] for a neural 'implementation' of multiplication.

**F.C.S.** In our model surface formation interacts with boundary formation, keeping the boundaries close to places of high brightness gradient, thereby avoiding the occasional shifting of edges due to higher level (stage III') cues, or the breaking up of corners, due to orientational competition. The brightness values of the image were used, instead of the On/Off- Off/On outputs, in order to compare our algorithm to others. A less important architectural, but significant practical modification we introduced was that we considered the B.C.S. neurons as being located between F.C.S. neurons, as in [30], shown also in Fig. 2(b). This facilitates the exploitation of the oriented line-process neurons by blocking the diffusion among two pixels only when there is an edge (close to) *perpendicular* to the line joining them. More formally, this can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}S(i,j) = \sum_{\theta} \nabla_{\theta^\perp} S(1 - (U^\theta)) \quad (8)$$

where, as in [25] $S(i,j)$ is always the subtracted quantity in $\nabla_{\theta^\perp} S$. The equations we used were modified to account for the discrete nature of the neighborhoods and the relative positions of the F.C.S./B.C.S. nodes, but the idea used is the same: block diffusion only across edges and not along them.

**Experimental Results and Model Evaluation**

The results of our model are shown at the finest scale in Fig. 3; for all the images the same set of parameters has been used, while modifying these by 10% did not result in significant changes in the system's performance. We used a small fraction of the parameters used e.g. in [14]; we did not manage to achieve the same results using the original B.C.S./F.C.S. model , even when optimising its parameters for every single image; extensive results and implementation details can be found in [31]. It should be noted that our model operates at specific scales, so it may not respond in the same way to stimuli of arbitrary size.

Even though our model compares favorably to Canny edge detection on these test images, we do not claim that it is superior to the state-of-the-art [4,7,25, 27] in edge detection. We consider it more important that the modifications we introduced resulted in a simpler and efficient biologically plausible model, that does not largely deviate from the original B.C.S./F.C.S. architecture.

## 4 Interpreting the Model in Computer Vision Terms

The B.C.S./F.C.S architecture clearly parallels the usage of line and surface processes by computer vision researchers, see e.g. [2,9,30]; it is therefore of interest to interpret the previously described network in computer vision terms. We start with the line process-related functions of the network and continue with the function of the network as a whole.

**Non-maximum Suppression:** Non-maximum suppression is commonly used for the purpose of deriving a clean and coherent edge map from fuzzy inputs like the outputs of spatially extended filters. A common technique for nonmaximum suppression is to take the local maximum of the filter responses in the gradient direction and set the others to zero, like in [4,7,24]; in [2,9] a penalty term punishes spatial configurations of broad edges, while in [27] fitting the surface with a parabola and using the curvature and distance to the peak was proposed. Keeping to biological plausibility, our system implements non-maximum suppression by an analog winner-take-all type network (7), that continuously suppresses the activations of less active neurons, allowing the stronger ones to stand out (see [31] for details).

**Perceptual Grouping and the Elastica Prior on Shapes:** For the goal of enhancing perceptually salient contours various techniques have been developed by computer vision researchers; a similar pattern of pixel interactions as that shown in Fig. 1(c) is used in the voting technique of [15], in the probabilistic formulations used in [27,28] edge saliency is propagated among processing nodes, while in [26] a criterion including the squared integral of the curvature is used for edge linking. The popular and simple hysteresis thresholding technique used in [4] can be seen as performing some sort of edge grouping, as well as the penalty term for line endings used in [2,9].

In our model the contour linking process is cooperating with the contour formation process, driving the latter to the more salient edges, while avoiding using initial hard decisions from an edge detector output. The shape of the lobes used to perform perceptual grouping, shown in Fig. 1(c) was introduced

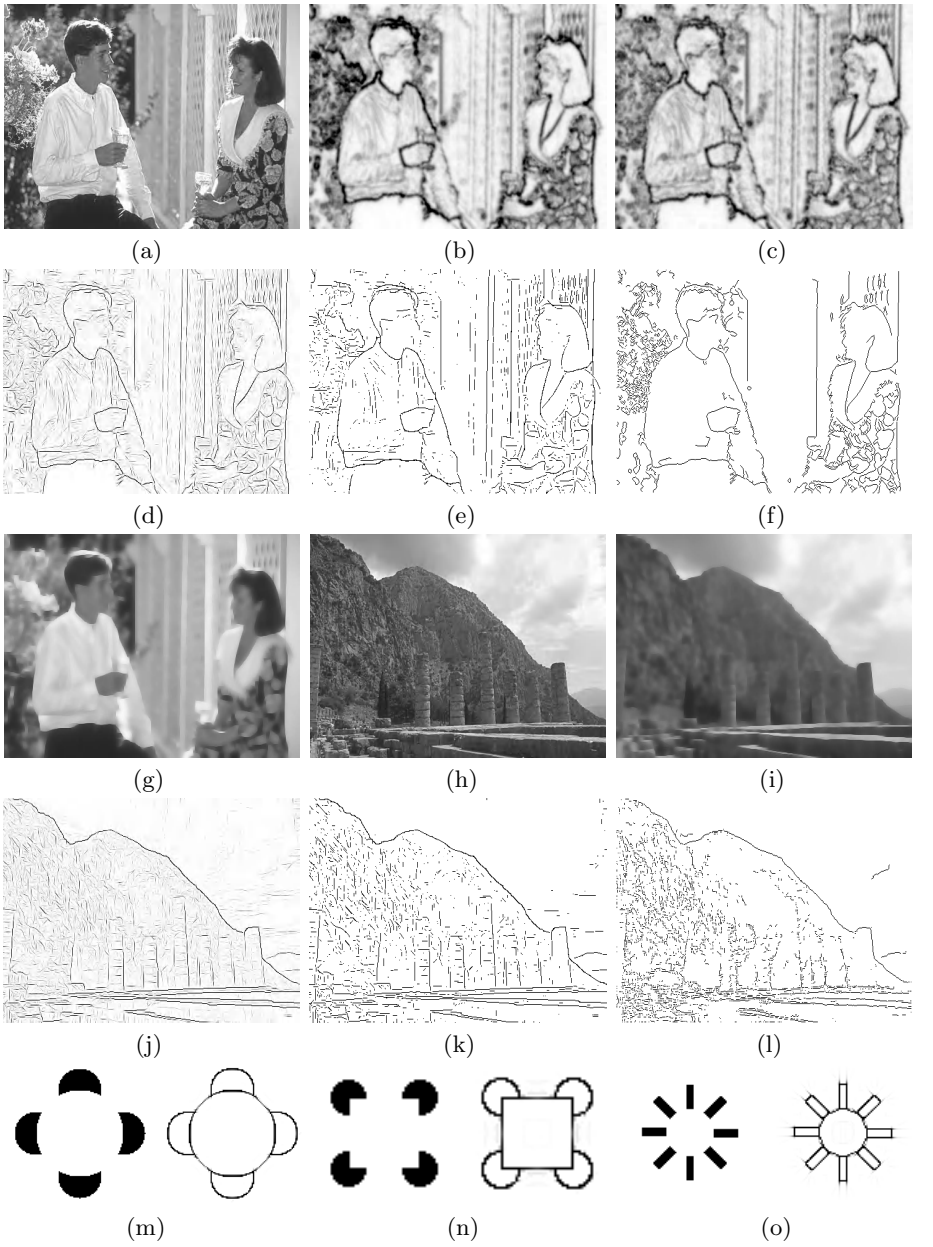



**Fig. 3.** Results of the proposed model: (a) input (b) orientation energy (c) normalized energy (d) line process at finest scale (e) thresholded results (f) Canny results with same number of pixels, $\sigma = 1, T_2 = .3, T_1 = .03$. (g) FCS outputs (h)-(l) same as (a),(g),(d)-(f) respectively, $\sigma = 3, T_2 = .1, T_1 = .01$ (m)-(o) results for 'Kanisza figures'.

by S. Grossberg and is now popular among researchers both in computer vision and biological vision [8,15,20,22,23]; this is natural since their shape, which favors low curvature contours, that occur frequently in our visual environment, enforces a reasonable prior on the contour formation process.

A very interesting link with the Elastica model of curves [21] is established in [28] where the shape of the lobes shown in Fig. 1(c) is related to the Elastica energy function $E(\Gamma) = \int_\Gamma 1 + a\kappa^2(s)\mathrm{d}s$, where $\kappa$ is the curvature of the curve $\Gamma$. If a particle starts at $[0, 0]$ with orientation 0 and the probability of its trajectory $\Gamma$ is $P(\Gamma) \sim \exp(-E(\Gamma))$, then the probability $P(x, y, \theta)$ of the particle passing through $x, y, \theta$ is very similar to the lobes in Fig. 1(c). Loosely speaking, one could say that the high-level feedback term calculated using the lobes shown in Fig. 1(a) is related to the posterior probability of a contour passing through a point, conditioned on its surroundings, using the prior model of Elastica on curves.

**A Variational Perspective:** The link between recurrent networks and Lyapunov functions [5,17] has been exploited previously [18,19,29,30] to devise neural networks that could solve variational problems in computer vision; we take the other direction, searching for a variational interpretation of the model we propose. Even though based on [5] one can find a Lyapunov function for the recurrent network described in the previous section, the integrals become messy and do not help intuition; we therefore consider the simplified version of (7):

$$\frac{\mathrm{d}V^{\theta,\sigma}}{\mathrm{d}t}[i,j] = -AV^{\theta,\sigma} + CI - D\sum_{\phi}\sum_{k,l} W^{\theta,\phi}_{[i,j]}[k,l]U^{\phi,\sigma}[k,l] \tag{9}$$

where instead of the synaptic interaction among neurons we use the common sum-of-inputs model. The main difference is that the absence of the multiplicative terms in the evolution equations leads faster to sharp decisions and hence more probably to local minima. For simplicity of notation we drop the scale index, considering every scale separately and treat temporarily the excitatory input $I$ as constant; a Lyapunov energy of the network is then [31]:

$$E = \sum_{i,j,\theta}[A\int_{1/2}^{U} g^{-1}(u)\mathrm{d}u - CIU] + D/2\sum_{i,j,\theta}\left[U^{\theta}[i,j]\sum_{\phi,k,l} W^{\theta,\phi}_{[i,j]}[k,l]U^{\phi}[k,l]\right] \tag{10}$$

In the above expression, $\int_{1/2}^{U} g^{-1}(u)\mathrm{d}u$ evaluates to $[U\ln(U) + (1-U)\ln(1-U)]/\beta + 1/2U - 1/4$ which consists of an entropy-like term, punishing 0/1 responses and $1/2U$ that generally punishes high responses; the first term is due to using a sigmoid transfer function and the second is due to shifting it by 1/2 to the right. The term $-IU$ lowers the cost of a high value of $U$, facilitating the emergence of an edge; among all $U$ of fixed magnitude ($\sum_i U(i)^2$) the one that minimizes $-\sum_i U_i I_i$ is the one that minimizes $\sum_i (U-I)^2$, thus explaining the product terms in (10) as enforcing the closeness of $U$ with $I$, without necessarily increasing $U$. The rightmost term in (10) can be expressed as [31]:

$$C(U) = \sum_{i,j,\theta}[G * U^{\theta}]^2 - b\sum_{i,j,\theta}[G^{\theta} * U^{\theta}(i,j)]^2 + \frac{1}{2}\sum_{i,j,\phi,\theta\neq\phi}\left[G^{\phi,\theta} * U^{\theta}\right]\left[G^{\theta,\phi} * U^{\phi}\right] .$$

The first two terms account for spatial sharpening of the edge map: $G$ is an elongated Gaussian which is a scaled by $\sqrt{2}$ in space version of the first term in (5) and can be interpreted as disfavoring broad features like $G * U^\theta$. $G^\theta$ is an elongated Gaussian that is a scaled by $\sqrt{2}$ in space version of the second filter in (5), so this term guarantees that an isolated in both space and orientation edge will not get inhibited, due to the other two terms. $C(U)$ thus consists of both a diffusive term, namely a penalty on $G*U^\theta$ that wipes away broad structures and a reactive term $-G^\theta * U^\theta$ that acts in favor of the emergence of isolated edges. The last term accounts for orientational sharpening and is expressed in terms of scaled in space copies of the Gaussian filters used for orientational competition.

Putting everything together requires incorporating the interaction with the surface process; using equation (8) and assuming $T^\theta$ constant, the expression

$$E = \sum_{i,j,\theta} c_4(1-U^\theta(i,j))|\nabla_\theta S|^2 - U^\theta[c_1 I^\theta + c_2 T^\theta + c_3 U^\theta_{\sigma+1}] + A\int_{1/2}^{U} g^{-1}(u)\mathrm{d}u + C(U)$$

can be shown to be [31] a Lyapunov function of the system, since by differentiating w.r.t $S$ and $U^\theta$ we get the evolution equations (8),(9) respectively. This functional can be seen as a more complex version of that introduced in [2], where a simple penalty term was used to enforce nonmaximum suppression and contour continuity to the anisotropic diffusion-derived line process.

## 5 Discussion

In this paper, motivated by ideas from biological vision, we proposed a simple and efficient model for low- and mid- level vision tasks which compares favorably to classical computer vision algorithms, using solely neural mechanisms. Its performance was demonstrated on both synthetical and real images, while its interpretation in computer vision terms has been presented, making the link with variational techniques. Extending our analysis to the whole FACADE model [11] of vision seems to be a promising future goal, which could result in a unified, biologically plausible model of computational mid-level vision.

## References

1. E. Adelson, and J. Bergen. Spatiotemporal energy models for the perception of motion *JOSA*, 2(2), 284–299, 1985.
2. M. Black, G. Sapiro, D. Marrimont, and D. Heeger. Robust anisotropic diffusion. *IEEE Trans. on Image Processing*, 7(3), 421–432, 1998.
3. M. Carandini and D. Heeger. Summation and division by neurons in visual cortex. *Science*, 264:1333–1336, 1994.
4. J. Canny, A computational approach to edge detection., *IEEE Trans. on PAMI*, 8(6):679–698, 1986.
5. M.A. Cohen and S. Grossberg. Absolute stability of global pattern formation and parallel memory storage by competitive neural networks. *IEEE Trans. on Systems, Man and Cybernetics*, 13(5):815–826, 1983.
6. J. Daugman. Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters *JOSA*, 2(7):160-169, 1985.

7. R. Deriche Using Canny's criteria to derive a recursively implemented optimal edge detector *IJCV*, 1(2):167–187, 1987.
8. D. Field, A. Hayes, and R. Hess. Contour integration by the human visual system: Evidence for a local 'association field'. *Vision Research*, 33(2):173–193, 1993.
9. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the bayesian treatment of images. *IEEE Trans. on PAMI*, 66(6):721–741, 1984.
10. S. Grossberg. *Neural Networks and Natural Intelligence*. MIT Press, 1988.
11. S. Grossberg. 3-d vision and figure-ground separation by visual cortex. *Perception and Psychophysics*, 55:48–121, 1994.
12. S. Grossberg and E. Mingolla. Neural dynamics of perceptual grouping: Textures, boundaries, and emergent segmentations. *Perc. & Psych.s*, 38:141–171, 1985.
13. S. Grossberg and E. Mingolla. Neural dynamics of surface perception: Boundary webs, illuminants, and shape from shading. *CVGIP*, 37:116–165, 1987.
14. S. Grossberg, E. Mingolla, and J. Williamson. Synthetic aperture radar processing by a multiple scale neural system for boundary and surface representation. *Neural Networks*, 8(7-8):1005–1028, 1995.
15. G. Guy and G. Medioni. Inferring global perceptual contours from local features. *IJCV*, 20(1):113–133, 1996.
16. F. Heitger and R.v.d. Heydt. A computational model of neural contour processing: Figure-ground segregation and illusory contours. *proc. ICCV*, pp. 32–40, 1993.
17. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. of the N.A.S. of USA*, 81:3088 – 3092, 1984.
18. C. Koch, J. Marroquin, and A. Yuille. Analog neuronal networks in early vision. *MIT AI lab Technical report 751*, 1985.
19. T. S. Lee. A bayesian framework for understanding texture segmentation in the primary visual cortex. *Vision Research*, 35:2643–2657, 1995.
20. Z. Li. Visual segmentation by contextual influences via intracortical interactions in primary visual cortex, *Network:Comput.Neur.Syst.*, 10:187–212, 1999.
21. D. Mumford. Elastica and computer vision. In Bajaj J., editor, *Algebraic Geometry and its applications*, pages 507–518. Springer Verlag, 1993.
22. H. Neumann and W. Sepp. Recurrent V1-V2 interaction in early visual boundary processing. *Biological Cybernetics*, 91:425–444, 1999.
23. P. Parent and S.W. Zucker. Trace inference, curvature consistency, and curve detection. *IEEE Trans. on PAMI*, 11(8):823–839, 1989.
24. P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *Proc. ICCV*, pp. 52–57, December 1990.
25. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. on PAMI*, 12(7):629–639, 1990.
26. A.Sha'ashua and S. Ullman. Structural saliency: the detection of globally salient structures using a locally connected network *Proc. ICCV*, pp. 321–327, 1988
27. X. Ren and J. Malik A probabilistic multi-scale model for contour completion based on image statistics. *Proc. ECCV*, pp. 312–327, 2002.
28. L.R. Williams and D.W. Jacobs. Stochastic completion fields: A neural model of illusory contour shape and salience. *Neural Computation*, 9(4):837–858, 1997.
29. A. Yuille. Energy functions for early vision and analog networks. *MIT A.I. lab technical report, 987*, 1987.
30. A. Yuille and D. Geiger A common framework for image segmentation. *IJCV*, 6:227–243, 1991.
31. I. Kokkinos, R. Deriche, O. Faugeras and P. Maragos Towards bridging the gap between biological and computational segmentation. *INRIA Research Report*, 2004.

# Appearance Based Qualitative Image Description for Object Class Recognition

Johan Thureson and Stefan Carlsson

Numerical Analysis and Computing Science, Royal Institute of Technology, (KTH), S-100 44 Stockholm, Sweden, {johant,stefanc}@nada.kth.se,
http://www.nada.kth.se/~stefanc

**Abstract.** The problem of recognizing classes of objects as opposed to special instances requires methods of comparing images that capture the variation within the class while they discriminate against objects outside the class. We present a simple method for image description based on histograms of qualitative shape indexes computed from the combination of triplets of sampled locations and gradient directions in the image. We demonstrate that this method indeed is able to capture variation within classes of objects and we apply it to the problem of recognizing four different categories from a large database. Using our descriptor on the whole image, containing varying degrees of background clutter, we obtain results for two of the objects that are superior to the best results published so far for this database. By cropping images manually we demonstrate that our method has a potential to handle also the other objects when supplied with an algorithm for searching the image. We argue that our method, based on qualitative image properties, capture the large range of variation that is typically encountered within an object class. This means that our method can be used on substantially larger patches of images than existing methods based on simpler criteria for evaluating image similarity.

**Keywords:** object recognition, shape, appearance

## 1 Introduction

Recognizing object classes and categories as opposed to specific instances, introduces the extra problem of within-class variation that will affect the appearance of the image. This is added to the standard problems of viewpoint variation, illumination etc. that induces variations in the image. The challenge is then to devise methods of assessing image similarity that capture these extra within-class variations while at the same time discriminate against objects of different classes. This problem has been attacked and produced interesting results, using quite standard methods of image representation relying heavily on advanced methods of learning and classification [2,3,6,8,9] These approaches are in general all based on the extraction of image information from a window covering part of the object, or in other words, representing only a fragment of the object. The size of

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 518–529, 2004.
© Springer-Verlag Berlin Heidelberg 2004

the fragments then effectively controls the within-class variation as different instances within a class are imaged. For a certain method of image representation and similarity assessment, there is in general an optimal size of the fragments in terms of discriminability, [2]. Increasing the size of the fragments beyond this size will decrease performance due to increased within-class variation that is not captured by the specific similarity criterion used. The fragment based approach can also be motivated from the fact that objects should be recognizable also in cases of occlusions, i.e. only a part of the object is available for recognition. It can therefore easily be motivated as a general approach to recognition. However, the main limitation of fragment based approaches still seems to lie in the fact that within class variation is not captured sufficiently by the similarity measures used. Ideally if we could use any fragment size, from very small up to fragments covering the whole object, we would improve the performance of algorithms for object class recognition.

Given any method to asses similarity of images, it's usefulness for *object class recognition* lies in it's ability to differentiate between classes. If similarity is measured normalized between 0 and 1.0 the ideal similarity measure would return 1.0 for objects in the same class and 0 for objects in different classes. In practise we have to be content with a gradually descending measure of similarity as images are gradually deformed. The important thing is that a sufficiently high value of similarity can be obtained as long as the two images represent the same object class.

## 2 Appearance vs. Shape Variation

Object shape is generally considered as a strong discriminating feature for object class recognition. By registering the appearance of an object in a window, object shape is only indirectly measured. Traditionally, object shape has been looked for in the gray value edges of the image, in general considered to coincide with 3D edges of the object. This however neglects the effects of object shape on the smoothly varying parts of the image which is ideally capture by appearance based methods. The split between edge based methods trying to capture object shape directly and appearance based methods giving an indirect indication of shape is unfortunate since they lead to quite different types of processing for recognition. Very little attention has been paid to the possibility of a unified representation of shape, accommodating both the direct shape induced variations of image edges and the indirect one's in the appearance of the image gray values. We will therefore investigate the use of qualitative statistical descriptors based on combinations of gradient directions in an image patch. The descriptors are based on the order type which encodes the qualitative relations between combinations of multiple locations and directions of gradients as described in [4], [5], [11] where it was used for correspondence computation in sparse edge images. The *histogram of order types* of an image patch is used as a descriptor for the patch. This descriptor then encodes statistical information about qualitative image structure

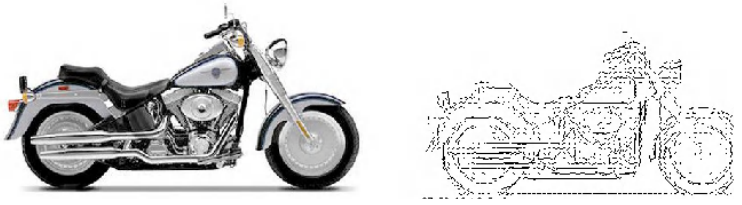


**Fig. 1.** The gradients of an image indirectly encode the shape of an object

and therefore potentially captures qualitative variations of images as displayed between the members of an object class

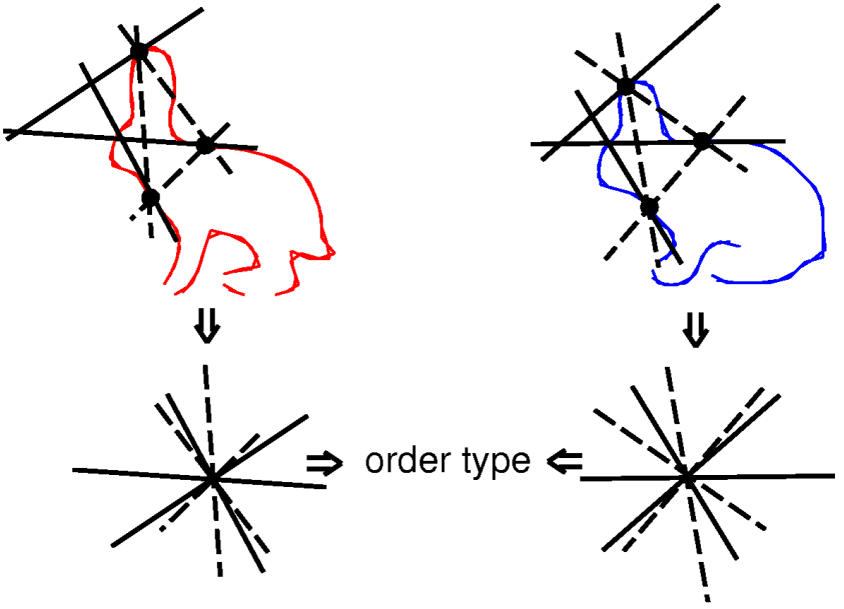



**Fig. 2.** Three points and their corresponding (orthogonal) gradient directions generates six lines. The angular order of these six lines defines the *order type* of the point-line collection. Corresponding points in qualitatively similar images in general generates the same order type.

If we take three points in an image together with the lines of gradient directions, (strictly speaking the directions orthogonal to the gradients), we can speak about the combinatorial structure of this combined set of points and lines. The three lines have an internal order by considering the angular orientations and the lines are ordered w.r.t. the points by considering the orientation of the lines relative to the points. The idea of *order type* of the combined set can be

easily captured by considering the three lines formed by connecting every pair of points together with the three gradient lines. The *angular order*, of these six lines then defines an index which defines the *order type* of the set. From fig. 2 we see that the order types for perceptually corresponding points in two similar shapes in general are equivalent. The unique assignment of points requires a canonical numbering of the three points for which we use the lowest point as number one and count clockwise. The order type is therefore an interesting candidate for a qualitative descriptor that stays invariant as long as a shape is deformed into a a perceptually similar shape. Order types as defined above are only strictly invariant to affine deformations. For more general types of deformations, invariance of the order type depends on the relative location of the three points. This means that if we consider the collection of all order types that can be computed by considering every triplet of points in an image that has a well defined gradient direction, we get a representation of the entire shape that has interesting invariance properties w.r.t. to smooth deformations that do not alter the shape too drastically. This is often the case between pairs of instances in an object category as noted very early by d'Arcy Thompson [10].

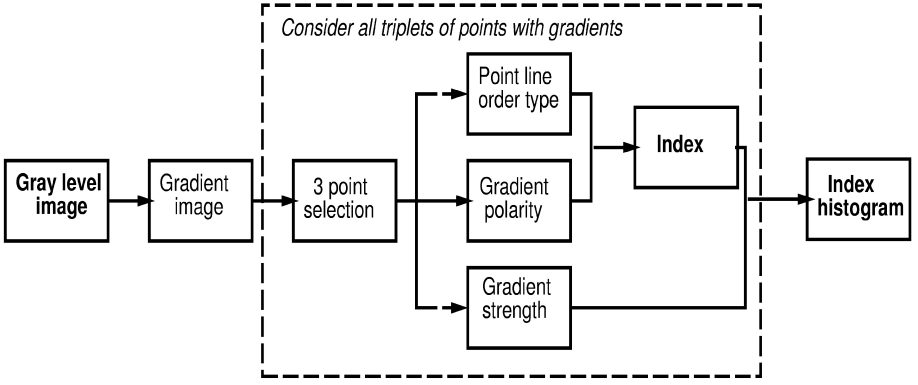



**Fig. 3.** Procedure for computing *weighted index histogram* from gray level image: A gradient detector is applied to the image. The gradients are thresholded and subsampled. All triplets of remaining gradients are selected and for each triplet we compute an index representing the joint qualitative structure of the three gradients based on order type and polarity. The index is given a weight depending on the strength of the gradients and a weighted histogram of index-occurrences is computed

The procedure for computing the *weighted index histogram* is illustrated by the block diagram of fig. 3. and can be summarized in the following points:

1. The image is smoothed with a Gaussian operator, a gradient detector is applied and it's output thresholded to give a preset total number of gradients
2. For *every combination of three gradients* we compute an *order type index* using the following procedure:

a) We choose the lowest point as number one and number the others in a clockwise ordering. The locations and angular directions of the three gradients are then denoted as $x_1, y_1, d_1, x_2, y_2, d_2, x_3, y_3, d_3$.
b) We compute the three directions $d_{12}, d_{13}$ and $d_{23}$ of the lines joining point pairs 12, 13, 23 respectively
c) The six numbers $d_1, d_2, d_3, d_{12}, d_{13}, d_{23}$ are ranked and the the permutation of the directions $1, 2, 3, 12, 13, 23$ is given a number denoted as the *order type index* for the three gradients
d) The polarity of each gradient is denoted which increases the index with a factor of $2^3 = 8$
e) The average strength of the three gradients is computed and used as a weight

3. A histogram of occurrence of the various order type + polarity indexes is computed by considering *all combinations of three gradients* in the image. Each index entry is weighted by the average strength of the three gradients used to compute the index

For the specific way of choosing the order type that we have , we get 243 distinct cases. By considering the polarity of the gradients, we get 1944 different indexes. The histogram of occupancy of these indexes as we choose all combinations of three points defines a simple shape descriptor for the image. In comparison, the same kind of histogram including a fourth point, was used in [4], [5], [11]. By fixating the fourth point a "shape context descriptor" [1] was computed and used for correspondence matching between shapes. Recognition was then based on the evaluation of this correspondence field. By using one single shape descriptor histogram for the whole image we get a comparatively less complex algorithm which allows us to consider every gradient in an image instead of just the selected directions after edge detection as in [4], [5], [11]. The histogram can be seen as a specific way to capture higher order statistics of image gradients where previously mainly first order has been used [7].

## 3 Comparing Images

Given the histogram vector based on qualitative image properties, comparisons between images can now be performed by simple normalized inner products. If the histogram vector captures qualitative image similarity we would expect images of a certain class to be close using this inner product while images of different classes should be further apart. A simple test of this property was made on two sets of images, motor bikes and faces, with about 15 examples in each class. A subset of those examples are shown in fig 4. Fig 5 shows histograms of inner products between all examples in two classes for the cases face-face, motor bike-motor bike and face-motor bike. This figure clearly demonstrates that pairs of images picked from the same class are definitively closer than pairs picked from different classes. The histograms vectors used are therefore able to capture similarity within a class and at the same time discriminate between different classes.

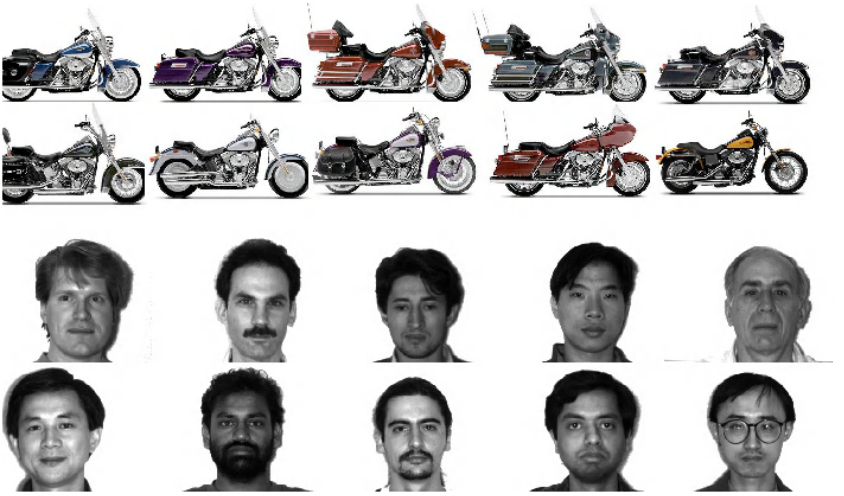


**Fig. 4.** Subset of images from two classes used for testing similarity measure

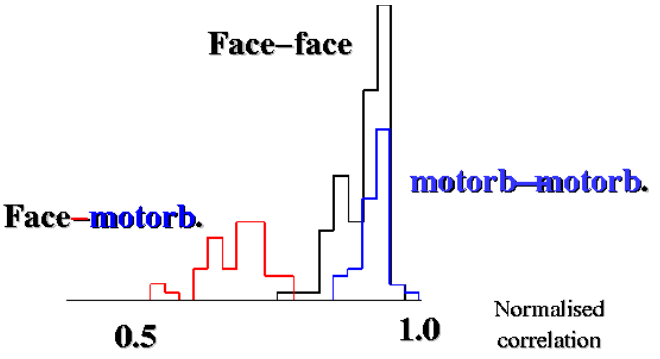



**Fig. 5.** Histograms of inner products of pairs of histogram vectors.

In another test we computed all inner products between the simple silhouette images of rabbits. Being silhouettes, the image differences are accounted for by shape differences. Using the inner products as a distance measure, we mapped the images in the rabbit set to positions in the plane such that the distance between two mapped rabbit examples corresponds to the distance given by the inner products as well as possible. This procedure is known as multi-dimensional scaling and from this it can be seen that the inner product distance corresponds quite well to a distance measure given by visual inspection of the rabbit examples. Note that there is a gradual transition of the shapes as we move from the upper to the lower part of this diagram

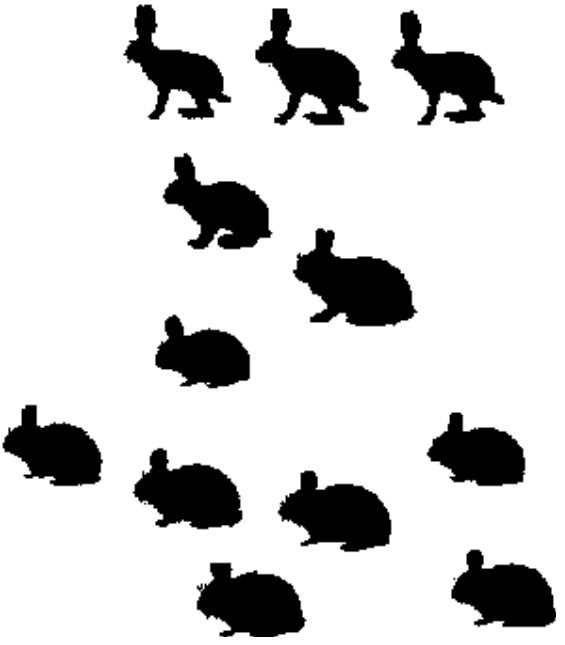


**Fig. 6.** Multidimensional scaling plot of shape examples using the inner product of histogram vectors to construct a distance measure

## 4 Object Classification: Experimental Results

We have tested our method on a database containing four categories of images: airplanes, cars, faces and motorcycles. A set of background images, that depicts varying objects, is also used. This database is the same as the one used in [6] downloaded from the image database of the Visual Geometry Group of the Robotics Research Group at Oxford at: `http://www.robots.ox.ac.uk/~vgg/data/`

Examples of images from the different sets are shown in figure 7, together with gradient images. The gradient images were obtained by gradient detection and subsampling after smoothing. By varying thresholds and subsampling rates we fixed the number of gradients to 500. Note that this occasionally implies that weak gradients from smooth areas in the background are detected. The fact that gradient strength is used in weighting the final index histograms serves to reduce the effect of these weak "noise" gradients.

Following the procedure in [6], we apply our method on each category in the following way. First the object set is split randomly into two sets of equal size. One set is for training; the other is for testing. The background set is used solely for testing. Thus there is no training of background images at this stage. For each image in the two test sets (object and background), the proximity to all the images in the training set is computed by inner multiplication of their index

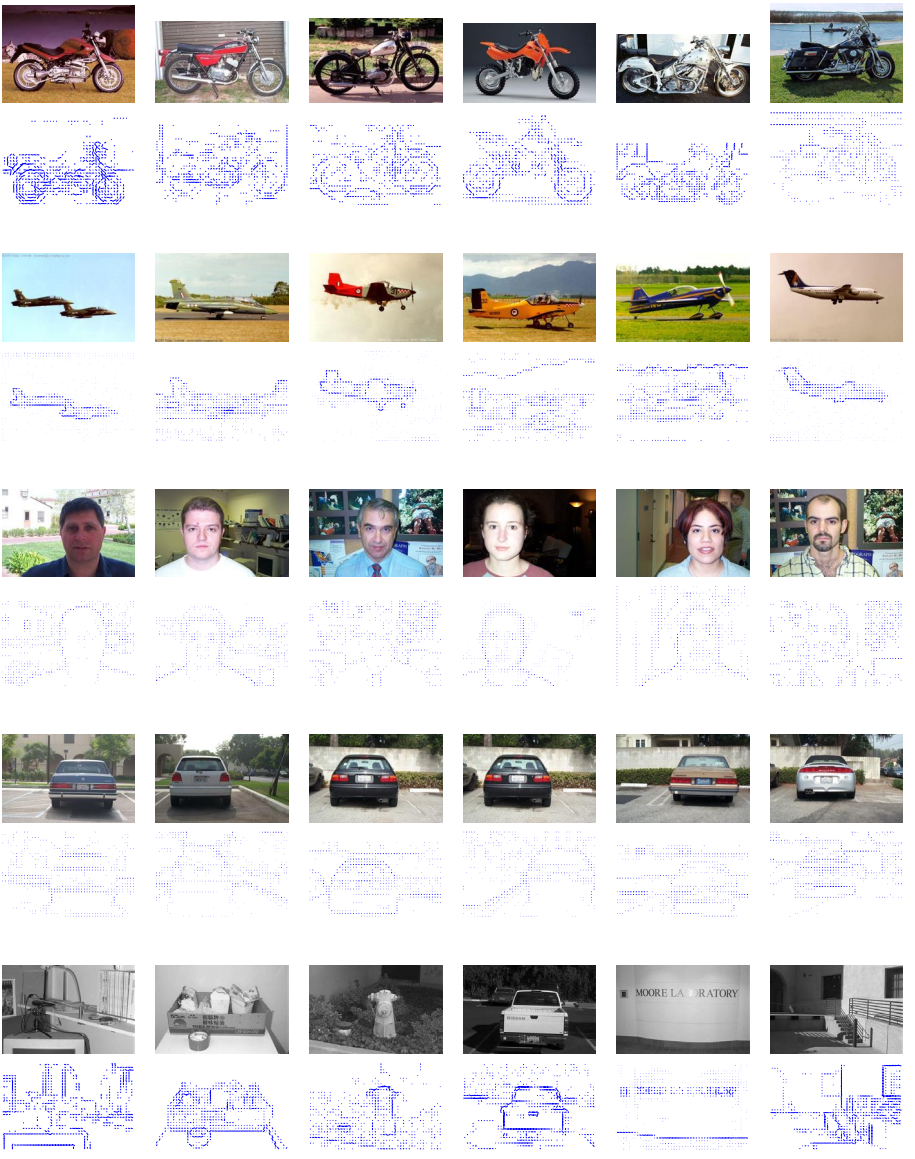


**Fig. 7.** Examples of images and gradient plots from the categories: airplanes, cars, faces, motorcycles, and background. These sets can be downloaded from the image database of the Visual Geometry Group of the Robotics Research Group at Oxford at: `http://www.robots.ox.ac.uk/~vgg/data/`

**Table 1.** Equal error rates for whole images.

| Dataset | #Images | NN | Edited NN | Ref:[6] | Ref:[12] | Ref:[13] |
|---|---|---|---|---|---|---|
| Faces | 450 | 81.8 | 83.1 | 96.4 | - | 94 |
| Airplanes | 1074 | 82.9 | 83.8 | 90.2 | 68 | - |
| Motorcycles | 826 | 93.0 | 93.2 | 92.6 | 84 | - |
| Cars | 126 | 87.3 | 90.2 | 84.8 | - | - |

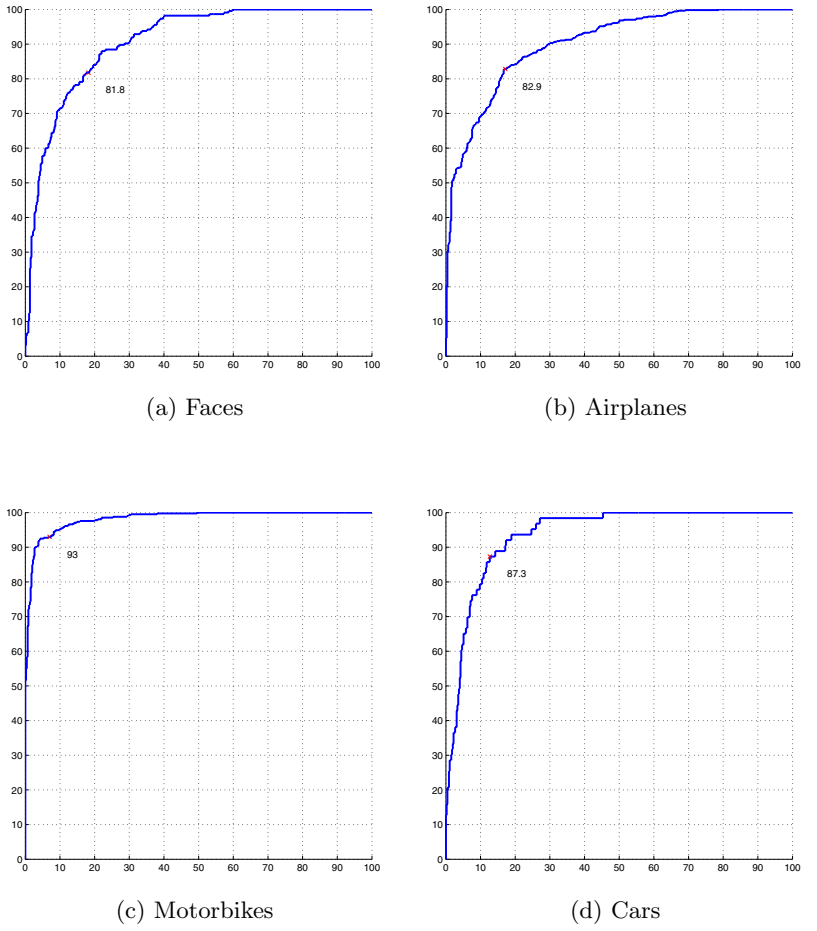
histograms. The highest value, corresponding to the nearest neighbor (NN) is found for each element in the test sets.

To decide whether an image belongs to a category or not, we compare its nearest neighbor proximity value to a threshold. The receiver-operating characteristic (ROC) curve is formed by varying this threshold. The curve shows false positives (fp) as a function of true positives (tp). In figure 8 we can see the ROC curve for faces, airplanes, motorcycles and cars. The equal error rate (EER) is the rate where tp=1-fp. In table 1 we can see the equal error rates for the four different categories compared to results for the same categories achieved by [6]. In the 'Edited' column, we can see the values for the edited nearest neighbor method, where training is performed with background images as well. Using this method the background database is split randomly into two halves; one training set and one test set. The proximities between all histograms of both the object and background training sets are computed. For each histogram of the object training set, the k (typically 3) nearest neighbors are found. If all of those are from the object training set, the object histogram is kept; otherwise it is removed from the training set. This achieves smoother decision boundaries.

In table 1 we see that for two of the categories (cars and motorcycles) we get better results than [6], whereas for the other two categories (faces and airplanes) we get worse results. The worse results are mainly due to gradients from the parts of the image surrounding the object. They cause a lot of noise in the histograms, which lowers the proximity value of the nearest neighbor for true positives. This is confirmed by the fact that when objects in the image are cropped, results improve. This is seen in table 2. An example of a whole image versus cropped can be seen in figure 9.

**Table 2.** Equal error rates for cropped images.

| Dataset | #Images | NN | Edited NN |
|---|---|---|---|
| Faces | 450 | 96.4 | 96.0 |
| Airplanes | 1074 | 86.7 | 89.2 |
| Motorcycles | 826 | 94.9 | 94.9 |
| Cars | 126 | 92.2 | 97.8 |




(a) Faces (b) Airplanes

(c) Motorbikes (d) Cars

**Fig. 8.** ROC curves with EER for faces, airplanes, motorcycles and cars.

## 5 Summary and Conclusions

We have presented a simple method for qualitative image description based on histograms of order type + polarity computed from the locations and gradient directions of triplets of points in an image. We have demonstrated that this descriptor captures the similarity of images of objects within a class while discriminating against images of other objects and arbitrary backgrounds. We have avoided the resort to small image patches or fragments that is necessary for most image similarity methods due to their inability to capture large variations within a class. The experimental results on recognition of images four objects in a large database was in two cases better than the best results reported so far [6] which

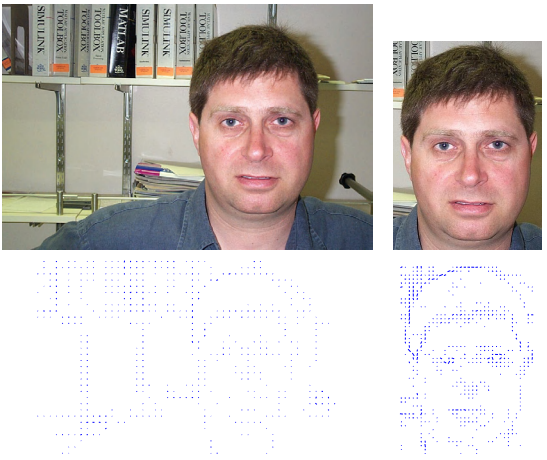



**Fig. 9.** Examples of whole and cropped face images and gradient plots.

uses a combination of patches to describe the object. In the two other cases, the amount of background clutter relative to the object was too large, giving results inferior to those in [6]. Using cropped images of these objects, our results are vastly improved demonstrating that by combining our algorithm with a search for the optimal size and location of a window covering the object, we should be able to handle these cases too.

The use of fragments can be motivated for reasons of recognition of occluded objects and for a potentially richer description of the object. In the future we will consider such descriptors where the size flexibility of our method will be of importance.

The use of qualitative image structure based on gradient locations and directions is essentially an appearance based method. It can however be tuned towards paying more attention to the strong structural properties of images which are undoubtly important in object classification, thereby forming a bridge between appearance and structure based methods for image recognition.

## References

1. Belongie S. and Malik J. Matching with Shape Contexts Proc. 8:th International Conference on Computer Vision (ICCV 2001)
2. Borenstein, E., Ullman, S., Class-Specific, Top-Down Segmentation, Proc. ECCV 2002
3. Burl, Weber, Perona A probabilistic approach to object recognition Proc. ECCV 1998 pp. 628 - 641
4. Carlsson. S, "Order Structure, Correspondence and Shape Based Categories", *International Workshop on Shape Contour and Grouping,* Torre Artale, Sicily, May 26-30 1998, Springer LNCS 1681 (1999)

5. Carlsson S. and Sullivan J. Action Recognition by Shape Matching to Key Frames Workshop on Models versus Exemplars in Computer Vision, Kauai, Hawaii, USA December 14th, 2001
6. Fergus, R., Perona, P., Zisserman, A., Object class recognition by unsupervised scale-invariant learning, CVPR03(II: 264-271). IEEE Abstract. IEEE Top Reference. 0307 BibRef
7. Schiele B. and Crowley J. Recognition without Correspondence using Multidimensional Receptive Field Histograms, IJCV(36), No. 1, January 2000, pp. 31-50.
8. Schneiderman, H., A Statistical Approach to 3D Object Detection Applied to Faces and Cars, Proc. CVPR 2000.
9. Sung, K.K., and Poggio, T., Example-Based Learning for View-Based Human Face Detection, PAMI(20), No. 1, January 1998, pp. 39-51.
10. Thompson, D'Arcy, "On growth and form" Dover 1917
11. Thureson J. and Carlsson S. Finding Object Categories in Cluttered Images Using Minimal Shape Prototypes, Proc. Scandinavian Conference on Image Analysis, (1122-1129). 2003
12. Weber, M. Unsupervised Learning of Models for Visual Object Class Recognition, PhD thesis, California Institute of Technology, Pasadena,CA, 2000
13. Weber, M., Welling, M., Perona, P., Unsupervised Learning of Models for Visual Object Class Recognition, Proc. ECCV 2000, pp 18 - 32 Springer LNCS 1842

# Consistency Conditions on the Medial Axis

Anthony Pollitt[1], Peter Giblin[1], and Benjamin Kimia[2]

[1] University of Liverpool, Liverpool L69 3BX, England
{a.j.pollitt,pjgiblin}@liv.ac.uk
[2] Brown University, Providence, RI, USA
kimia@lems.brown.edu

**Abstract.** The medial axis in 3D consists of 2D sheets, meeting in 1D curves and special points. In this paper we investigate the consistency conditions which must hold on a collection of sheets meeting in curves and points in order that they could be the medial axis of a smooth surface.

## 1 Introduction

The Medial Axis (MA) was introduced by Blum as a representation of shape [1] in the early 1960s. Its intuitive appeal has since captured the imagination of researchers in fields where the representation of shape plays a key role, including object recognition, path planning, shape design and modelling, shape averaging, morphing shape, and meshing unorganized point samples, among many others. In many of these applications, the ability to recover the original shape from the medial axis is highly significant. In this paper we give details of constraints on the medial axis at special points, which will effect a more precise reconstruction of the original shape.

In the context of object recognition, while the early literature on the use of the medial axis is dominated by methods for computing its geometric trace, the use of a graph-based representation of the medial axis has enabled robust recognition in the face of instabilities [15,19,23]. The definition of a shape similarity metric as mediated by the skeletal graph depends in a significant way on the ability to reconstruct the shape from its medial axis [6,18]. [1] This requires a knowledge of the constraints. Other areas where this is important include stochastic shape [13,11]; the averaging of a group of shapes, examples of which occur in medical imaging [8]; the matching of regions using 'deformable shape loci' model [16]; and industrial design [3]. Also, the field of ergonomics uses average human form measurements to optimize the interaction of people with products. The reconstruction of shape from the medial axis is also critical in robotics, where the medial axis of the free space is used for path planning [9,10].

In 2D, the medial axis of a generic planar shape consists of smooth branches with endpoints, the branches meeting in threes at points variously known as

[1] in particular in using the shock graph [18] where the graph-based geometric trace is augmented with a qualitative motion of flow; the dynamically defined shock graph is more consistent with perceptual categories of shape [20].

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 530–541, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Y-junctions, triple intersections or $A_1^3$ points; see for example [7]. Each point of the medial axis is the centre of a 'maximal' circle which is tangent to the shape boundary in at least two points, or, for endpoints of the medial axis, is tangent to the shape boundary at an extremum of curvature. Conversely, given a smooth branch $\gamma$ and a 'radius' function $r$ we can reconstruct, at any rate locally, the two corresponding parts $\gamma^+$ and $\gamma^-$ of the outer boundary as an envelope of circles, centred on the smooth branch and of radius $r$. (See for example [6].)

However, given a connected set of smooth branches with ends and Y-junctions it is far from clear that this can be the medial axis of a shape with a smooth boundary. Even considering the local situation, it is not clear that from an arbitrary Y-junction furnished with three radius functions (agreeing at the junction point) we can expect to recover a smooth shape. In [6] it was shown that at the Y-junction there are constraints on the geometry of the three branches and on the 'dynamics', that is the derivatives of the three radius functions. One of the simplest of these constraints is

$$\frac{\kappa_1}{\sin\phi_1} + \frac{\kappa_2}{\sin\phi_2} + \frac{\kappa_3}{\sin\phi_3} = 0 \tag{1}$$

where the $\kappa_i$ are the curvatures of the medial axis branches at the junction and the $\phi_i$ can be interpreted here as the angles between the branches ($\phi_1$ is the angle between the branches with curvatures $\kappa_2, \kappa_3$, etc.). (See also [5,4,21,22] for work on the relationship between the curvatures of the medial axis and the boundary.)

The constraints arise because close to a Y-junction there are two ways to use the medial axis to construct each piece of the shape boundary. Given two smooth medial axis branches $\gamma_1, \gamma_2$, close to a Y-junction, and choosing orientations suitably, we reconstruct $\gamma_1^{\pm}$ and $\gamma_2^{\pm}$. In this case, $\gamma_1^+$ must agree with $\gamma_2^-$ at the point where they meet. (See Fig. 1, left.)

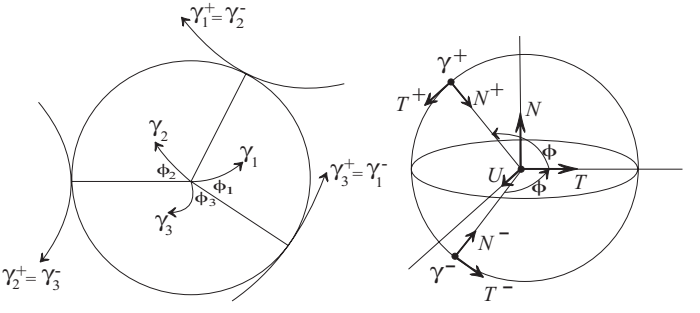



**Fig. 1.** Left: the $A_1^3$ case in 2D. Right: the 3D situation of the $A_1^2$ case

In this paper we take the step from 2D into 3D. The medial axis of a 3D region [2,7,12], or of the smooth surface $S$ bounding such a region, is defined as the closure of the locus of centres of spheres tangent to $S$ in at least two places,

only *maximal* spheres being considered, that is spheres whose radius equals the shortest distance from the centre to $S$. Using the standard notation of Fig. 2, the medial axis in 3D has sheets of $A_1^2$ points (ordinary tangency at two points and the centres of the spheres are maximal); curves of $A_1^3$ points (triple tangency) and $A_3$ points (edge of the medial axis where spheres have contact along a ridge); and these curves end in $A_1^4$ points (quadruple tangency) and $A_1A_3$ (or fin) points.

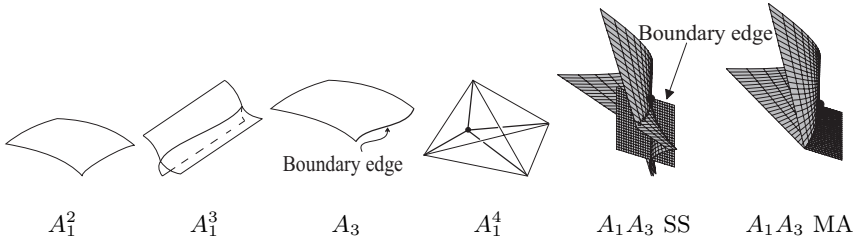



**Fig. 2.** The local forms of the 3D medial axis in the generic case. Thus $A_1^2$ is a smooth sheet, $A_1^3$ is a Y-junction curve, $A_3$ is a boundary edge of a smooth sheet, $A_1^4$ is a point where four $A_1^3$ curves lying on six sheets meet. Finally the '$A_1A_3$ SS' figure shows the 'symmetry set', consisting of a swallowtail surface and a smooth sheet with boundary; this is truncated as shown in the $A_1A_3$ medial axis figure. The large dot marks where the boundary ($A_3$) edge and the Y-junction ($A_1^3$) curve end at the $A_1A_3$ point—also called a 'fin point'—itself.

This paper examines the constraints on the medial axis from a theoretical perspective; applications will follow. The results on the 3D medial axis are new; see Prop. 1, Prop. 2, (9) and Prop. 5 for the Y-junction ($A_1^3$) case. See Prop. 3 and (13) for $A_3$, Prop. 4 for $A_1^4$ and Prop. 6 for $A_1A_3$. See [17] for full details.

**Acknowledgements.** Pollitt and Giblin acknowledge the support of the British Research Council EPSRC; Giblin and Kimia acknowledge the support of NSF. The authors have also benefited greatly from conversations with Professor J. Damon.

## 2 The Medial Axis in Three Dimensions

We consider a smooth ($A_1^2$) medial axis sheet $\gamma$ and its corresponding local boundary surfaces $\gamma^+$ and $\gamma^-$ in 3D and quote some connected definitions and formulae from [7]. Thus spheres (radius $r$) centred on $\gamma$ are tangent to $\gamma^\pm$. We have:

$$\gamma^\pm = \gamma - rN^\pm \ , \tag{2}$$

where $N^\pm$ is the unit normal to the boundary surface, oriented towards the centre of the bitangent sphere (i.e. tangent in at least two points). We shall assume that near a point of interest on $\gamma$ the gradient of $r$ is non-zero, and use the coordinate system given by the lines $r$ = constant, parametrized by $t$, say

and referred to as '$t$-curves', and the gradient lines of $r$, parametrized by $r$ and referred to as '$r$-curves'. We can fix the $(r, t)$ coordinate system close to $(r_0, 0)$ say, by taking $t$ to be arclength along the $t$-curve $r = r_0$. We denote partial derivatives by suffices.

The unit vector $T$ is defined to be parallel to $\gamma_r$, with $N$ a unit normal vector to the medial axis and taking $U = N \times T$ makes an orthonormal triad $[T, U, N]$. The velocity $v$ is defined by:

$$\gamma_r = vT, \quad \gamma_t = wU \ , \tag{3}$$

and $w$ a function which satisfies $w(r_0, t) = 1$ for all $t$. See Fig. 1, right. The convention of taking the boundary point $\gamma^+$ to be on the '$+N$' side of $T$ and denoting $\phi$ to be the angle $(0 < \phi < \pi)$ turned anticlockwise from $T$ to $-N^+$, in the plane oriented by $T, N$, means that

$$N^\pm = -\cos\phi T \mp \sin\phi N \ , \ T^\pm = \mp \sin\phi T + \cos\phi N \ , \text{ and } \cos\phi = -\frac{1}{v} \ . \tag{4}$$

Here the unit vector $T^\pm$ is tangent to $\gamma^\pm$ and parallel to $U \times N^\pm$. The equations (3), (4) hold for all points of $\gamma$.

The second order derivatives of $\gamma$ at $\gamma(r_0, 0)$ are as follows.

$$\gamma_{rr} = aT - va^t U + v^2 \kappa^r N, \ \gamma_{rt} = a^t T + a^* U - v\tau^t N, \ \gamma_{tt} = -\frac{a^*}{v} T + \kappa^t N \ , \tag{5}$$

where $a$, $a^t$, $a^*$ are defined everywhere on $\gamma$ respectively as $v_r$, $v_t$, $w_r$ and $\kappa^r$, $\kappa^t$, and $\tau^t$ are respectively the *normal curvature* in the direction of the $r$-curve, the *normal curvature* in the direction of the $t$-curve, and the *geodesic torsion* in the direction of the $t$-curve.

## 3 The $A_1^3$ (Y-junction Curve) Case

We consider the case when the medial axis of the boundary surface $S$ is locally three sheets, $\gamma_1$, $\gamma_2$, $\gamma_3$, say intersecting in a space curve, the $A_1^3$, or Y-junction, curve. Points of this curve are called Y-junction points and are centres of *tritangent* spheres, i.e. spheres tangent to the boundary in three points. The local boundary surfaces associated to $\gamma_i$ are $\gamma_i^\pm$. There are six associated boundary surfaces and each coincides with one, and only one, of the others, so there are three distinct boundary surfaces. Let the identifications be $\gamma_i^+ = \gamma_{i+1}^-$, i.e. $\gamma_1^+ = \gamma_2^-$, $\gamma_2^+ = \gamma_3^-$, $\gamma_3^+ = \gamma_1^-$. This is analogous to the Y-junction case in 2D – see Fig. 1, left. Making such identifications has consequences for the geometry and dynamics of the medial sheets $\gamma_1$, $\gamma_2$, $\gamma_3$. We obtain constraints by imposing conditions which use successively more derivatives. These constraints can be expressed in terms of two coordinate systems, which are as follows.

**First Coordinate System.** On each medial sheet $\gamma_i$ there is a radius function $r_i$, which can be used to set up the 'grad$(r_i)$-coordinate system' as defined in Sect. 2 for a single $A_1^2$ sheet. Hence on $\gamma_i$ we have

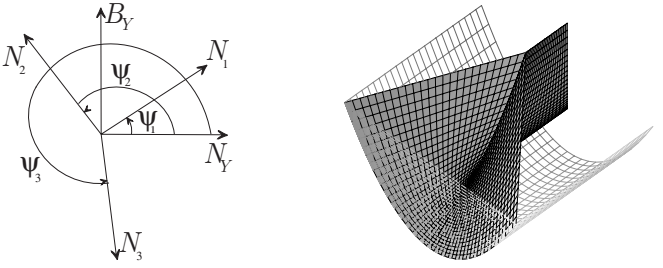



**Fig. 3.** Left: this is in the plane of $N_Y$, $B_Y$, respectively the principal normal and the binormal to the Y-junction curve. The unit normals $N_i$ to the medial sheets $\gamma_i$ and the angles $\psi_i$ between $N_Y$ and $N_i$ are shown. Right: an example of a boundary surface consisting of a parabolic cylinder with an end and its medial axis near to a Y-junction curve. The boundary is in wireframe, the medial axis is shaded

- $T_i$, $N_i$ – respectively the unit tangent to the $r_i$-curve, unit normal to $\gamma_i$;
- $U_i = N_i \times T_i$, the angle $\phi_i$ from $T_i$ to $-N_i^+$; velocity $v_i = -1/\cos\phi_i$;
- three accelerations: $a_i$, $a_i^t$, $a_i^*$;
- three 'geometries': $\kappa_i^r$, $\kappa_i^t$, $\tau_i^t$ – respectively the *normal curvature* in the direction of the $r_i$-curve, and the *normal curvature, geodesic torsion* in the direction of the $t_i$-curve.

**Second Coordinate System.** This coordinate system is specially adapted to the Y-junction case. We set up a local Frenet frame (which will exist for generic Y-junction curves, where the curvature is non-zero) at a point $\gamma_i(r_i = r_0, t_i = 0)$ of the Y-junction curve; thus we define the following.

- $T_Y$, $N_Y$, $B_Y$, $\kappa$, $\tau$ are respectively the unit tangent, principal normal, binormal, curvature and torsion of the Y-junction curve;
- $\alpha_i$, $\psi_i$ are respectively the angle from $T_Y$ to $T_i$, the angle from $N_Y$ to $N_i$. Choose $\psi_i$ such that $0 \leq \psi_1 < \psi_2 < \psi_3 < 2\pi$ (see Fig. 3, left);
- $r$ is the radius of the tritangent sphere centred on a Y-junction point;
- $\phi_Y$ is the angle between $T_Y$ and $-N_i^{\pm}$. Choose $\phi_Y$ to be obtuse, corresponding to $T_Y$ being in the direction of $r$ increasing along the Y-junction curve;
- $\kappa_i^W$ is the normal curvature of $\gamma_i$ in the direction of $W_i = N_i \times T_Y$;
- $'$ ('prime') denotes differentiation with respect to arclength along the Y-junction curve.

The simplest constraints use the first coordinate system and are as follows (see Sect. 7 for details).

**Proposition 1.** *At $\gamma_i(r_0, 0)$ on the Y-junction curve the following holds.*

$$\sum_{i=1}^{3}\left(\kappa_i^t \sin\phi_i + \frac{\kappa_i^r}{\sin\phi_i}\right) = 0 \ , \ \sum_{i=1}^{3}\left(\frac{a_i^* \kappa_i^r v_i - a_i \kappa_i^t - 2a_i^t \tau_i^t v_i}{v_i^3 \sin\phi_i}\right) = 0 \ . \quad (6)$$

The equation on the left of (6) is a constraint *only on the geometry ($\kappa_i^r$, $\kappa_i^t$) of the medial axis; it does not involve the dynamics*. Compare this with the simplest constraint (1) in the 2D case.

The identifications $\gamma_i^+ = \gamma_{i+1}^-$ on the Y-junction curve and (2) mean we have $N_i^+ = N_{i+1}^-$. The normals $N_i^\pm$ can be expressed in terms of $\phi_Y$, $\alpha_i$, $\phi_i$ and $\psi_i$ using the second coordinate system. Then the following can be proved.

**Proposition 2.** *At all points of the Y-junction curve, $N_i^+ = N_{i+1}^-$ for $i = 1 \ldots 3$ if and only if all of the following hold.*

$$\cos\phi_i = \epsilon_i \cos\phi_Y \sqrt{1 + \tan^2\phi_Y \cos^2\theta_i} \ , \ \sin\phi_i = \sin\phi_Y \sin\theta_i \ , \quad (7)$$

$$\cos\alpha_i = \frac{\epsilon_i}{\sqrt{1 + \tan^2\phi_Y \cos^2\theta_i}} \ , \ \sin\alpha_i = \frac{-\tan\phi_Y |\cos\theta_i|}{\sqrt{1 + \tan^2\phi_Y \cos^2\theta_i}} \ , \quad (8)$$

*where $\theta_i = \psi_{i+2} - \psi_{i+1}$ and $\epsilon_i = \mathrm{sign}(\cos\theta_i)$.*

At the second order level of derivatives we involve the geometry of the Y-junction curve to get another constraint. We can obtain expressions for $\kappa_i^r$, $\kappa_i^t$, and $\tau_i^t$ at $\gamma_i(r_0, 0)$ in terms of $\kappa$, $\tau$, $\kappa_i^W$, $\psi_i'$, $\phi_Y$, and $\psi_{1,2,3}$. Using these expressions it can be shown that the basic constraint on the left of (6) is the same as the following.

$$\sum_{i=1}^{3}\left(\frac{\kappa_i^W \sin^2\phi_Y - 2(\tau + \psi_i')\cos\phi_Y \sin\phi_Y \cos\theta_i + \kappa\cos^2\phi_Y \cos\psi_i}{\sin\theta_i}\right) = 0 \quad (9)$$

For more on the Y-junction case in general, see Sect. 7. Now we consider an example of these constraints.

**Example.** Suppose the boundary surface to be a parabolic gutter given by $z = by^2$ and the plane $x = p$ for $b$, $p$ constants. Then a sphere whose centre lies on the $A_1^3$ curve has two points of contact with $z = by^2$ and one with $x = p$. See Fig. 3, right where $b = 1$, $p = 0.6$. The medial axis can then be calculated explicitly in terms of $(x, y)$ for $y > 0$ and where $(x, y, by^2)$ is a point on the parabolic gutter and we get

$$\gamma_1(x, y) = \left(x, 0, by^2 + \frac{1}{2b}\right) \ , \ \gamma_2(x, y) = \left(x, -y(1 - 2b\lambda), by^2 + \lambda\right) \ , \quad (10)$$

$$\gamma_3(x, y) = \left(x, y(1 - 2b\lambda), by^2 + \lambda\right), \text{ where } \lambda = \frac{p - x}{\sqrt{1 + 4b^2y^2}} \ , \quad (11)$$

$$r_1(x, y) = \frac{1}{2b}\sqrt{1 + 4b^2y^2}, \ r_2(x, y) = p - x, \ r_3(x, y) = p - x \ . \quad (12)$$

The $A_1^3$ curve (which we shall call $C$) is given by the transversal intersection of $\gamma_1$, $\gamma_2$, and $\gamma_3$. (So $1 - 2b\lambda = 0$, giving $x$ in terms of $y$.)

From these parametrizations we can calculate explicitly the terms that appear in the constraints above. We can calculate formulae for $\psi_i$, $\alpha_i$, $\phi_i$, $\phi_Y$ in terms of $y$, and so we can verify (7) and (8) from Prop. 2 for this example. In a similar way we can check that (9), which is the same as the equation on the left of (6) from Prop. 1, holds in this example.

## 4 The $A_3$ Case

The $A_3$ case is when $\gamma^+(r_0, 0) = \gamma^-(r_0, 0)$ (consider Fig. 1, right – this corresponds to $\phi = 0$ or $\pi$). The corresponding points of the medial axis $\gamma$ are edge points, for which the velocity $v$ satisfies $v^2 = 1$ (see (2), (4)). We discovered the condition for the boundary surface to be smooth at an edge point $\gamma(r_0, 0)$, which is as follows.

$$\frac{a}{v} - r_0 \left( \frac{aa^*}{v} + (a^t)^2 \right) \neq 0 \tag{13}$$

In the case of the medial axis $\gamma$ being a general cylinder parametrized as $\gamma(u, z) = \delta(u) + z(0, 0, 1) = (X(u), Y(u), z)$ for $\delta$ a unit speed curve, the condition for smoothness of the boundary at a point of the $z$-axis where $v^2 = r_u^2 + r_z^2 = 1$ and $r_0$ is $r$ at this point is

$$r_0(r_{uu}r_{zz} - r_{uz}^2) - (r_u^2 r_{uu} + 2r_u r_z r_{uz} + r_z^2 r_{zz}) \neq 0 \ . \tag{14}$$

**Connection Between Curvatures.** A parametrization of the medial axis near to an edge point can be obtained by taking the associated boundary surface $S$ in Monge form near to an $A_3$ point. Let $S$ be given by $(x, y, f(x, y))$ where

$$\begin{array}{l} f(x,y) = \frac{1}{2}(\kappa_1 x^2 + \kappa_2 y^2) + (b_1 x^2 y + b_2 x y^2 + b_3 y^3) \\ +(c_0 x^4 + c_1 x^3 y + \cdots) + (d_0 x^5 + d_1 x^4 y + \cdots) + \cdots \end{array} \tag{15}$$

(Note $b_0 = 0$ – this corresponds to $A_3$ along the $x$-direction.) Explicit calculations give the following.

**Proposition 3.** *The limiting value (if it exists) of the Gauss curvature $K$ (which is the product of the principal curvatures on the medial axis) as $x, y \to 0$, i.e. as we tend towards the edge point is*

$$K = \frac{4(4b_2 d_0 - c_1^2)\kappa_1^4}{((\kappa_1 - \kappa_2)(\kappa_1^3 - 8c_0) - 4b_1^2)^2} \ . \tag{16}$$

The denominator of the right hand side of (16) is only zero if $\mathbf{0}$ is an $A_4$ point, which we assume is not true. Equation (16) is analogous to the $A_3$ case in 2D (see (92) of Lemma 8 from [6].) The limiting value of $K$ as in (16) depends on the 5-jet of the boundary in three dimensions. Equation (16) gives a criterion for $K = 0$ on the medial axis. We can also express $K$ in terms of derivatives of the principal curvatures $\kappa_1$ and $\kappa_2$ at $(x, y) = (0, 0)$ on the boundary (see [17]). This is also analogous to the situation in 2D.

Now consider the reconstruction of the boundary near to a ridge curve. Assume we are given the edge as a space curve, $r$ along it, the tangent planes to $\gamma$, and the angle between the tangent to the edge and grad($r$) at a point of the edge. Then it can be shown that we obtain the ridge curve, the tangent planes to the boundary along the ridge, the principal curvatures and principal directions of the boundary along the ridge. Compare this with (16) from Prop. 3, where very high order information about the boundary at a ridge point is required to determine second order information on the medial axis.

**Local Maximum or Minimum of $r$ along a Ridge Curve.** It can be shown that the derivative of $r$ with respect to arclength along the ridge is equal to zero at $x = y = 0$ if and only if $b_1 = 0$. Hence, using [14, pp.144], $r$ having a local maximum or minimum on the edge is the same as $\kappa_1$ *having a critical point on* $S$.

## 5 The $A_1^4$ Case

When a sphere is tangent to the boundary surface $S$ in four distinct points, the centre of this sphere is called an $A_1^4$ point. On the medial axis this corresponds to six sheets $\gamma_i$, for $i = 1 \ldots 6$, intersecting in a point, which is the $A_1^4$ point (see Fig. 2). At this point, four Y-junction curves intersect. Each medial axis sheet contains two Y-junction curves. We can obtain constraints on the medial axis at an $A_1^4$ point similar to those in Sect. 3, but these are complicated and so are omitted (see [17]). Here we give a result about the reconstruction of the four boundary points.

**Proposition 4 (Reconstruction at $A_1^4$ Points).** *At an $A_1^4$ point, the four contact points between the boundary surface $S$ and a maximal sphere are determined by the directed tangents to the four Y-junction curves, pointing into the medial axis, and the radius of the sphere.*

## 6 Radial Shape Operator

Now we shall give details of the calculations which gave the constraints of Sect. 3 and more complicated constraints which follow in the Y-junction case (Sect. 7). This section contains results connecting the geometry and dynamics of a medial axis sheet with the two corresponding boundary sheets, which are of use in Sect. 7. These results were obtained by using Damon's *Radial Shape Operator* $S_{\text{rad}}$ [4] of an $n$-dimensional medial axis $M \subset \mathbb{R}^{n+1}$ with associated boundary $B$. For $R$ a multivalued radial vector field from points of $M$ to the corresponding points of $B$, with $R = rR_1$, where $R_1$ is a unit vector field and $r$ is the radius function, the boundary $B = \{x + R(x) : x \in M \text{ all values of } R\}$. (In [4] $U$ is used for our $R$, but for us $U$ is already in use.) Also, in a neighbourhood of a point $x_0 \in M$ with a single smooth choice of value for $R$, let the radial map be given by $\psi_1(x) = x + R(x)$.

**Definition 1.** *([4, §1].) At a non-edge point $x_0$ of $M$ with a smooth value of $R$, let*

$$S_{\mathrm{rad}}(v) = -\mathrm{proj}_R\left(\frac{\partial R_1}{\partial v}\right) \tag{17}$$

*for $v \in T_{x_0}M$. Here $\mathrm{proj}_R$ denotes projection onto $T_{x_0}M$ along $R$. Also, $\frac{\partial R_1}{\partial v}$ is the covariant derivative of $R_1$ in the direction of $v$. The principal radial curvatures $\kappa_{ri}$ are the eigenvalues of $S_{\mathrm{rad}}$. The corresponding eigenvectors are the principal radial directions. Let $d\psi_t(v_i) = \frac{\partial \psi_t}{\partial v_i}$.*

Now we will specialize to consider a smooth medial axis $M \subset \mathbb{R}^3$. We assume the boundary is smooth and take $-N^{\pm}$ for $R_1$. We shall choose a basis $\mathbf{v}$ for $T_{x_0}M$ and obtain the radial shape operator matrix (denoted $S^{\pm}_{\mathbf{v}}$) with respect to this basis.

**Lemma 1.** *([4, Theorem 3.2].) For a smooth point $x_0$ of $M$, let $x_0' = \psi_1(x_0)$, and $\mathbf{v}'$ be the image of $\mathbf{v}$ for a basis $\{V_1, V_2\}$. Then there is a bijection between the principal curvatures $\kappa_i$ of $B$ at $x_0'$ and the principal radial curvatures $\kappa_{ri}$ of $M$ at $x_0$ (counted with multiplicities) given by $\kappa_i = \kappa_{ri}/(1 - r\kappa_{ri})$. The principal radial directions corresponding to $\kappa_{ri}$ are mapped by $d\psi_1$ to the principal directions corresponding to $\kappa_i$.*

We shall choose $\{V_1, V_2\} = \{T, U\}$ to be the basis for $S_{rad}$ of $\gamma$ at $\gamma(r_0, 0)$. Then we get

$$S^{\pm}_{\mathbf{v}} = \begin{pmatrix} s^{\pm}_{11} & s^{\pm}_{12} \\ s^{\pm}_{21} & s^{\pm}_{22} \end{pmatrix} = \begin{pmatrix} -\frac{a}{v^3 \sin^2\phi} \pm \frac{\kappa^r}{\sin\phi}\left(-\frac{a^t}{v^2} \mp \tau^t \sin\phi\right)\frac{1}{\sin^2\phi} \\ -\frac{a^t}{v^2} \mp \tau^t \sin\phi \end{pmatrix} \quad \begin{matrix} \\ \frac{a^*}{v^2} \pm \kappa^t \sin\phi \end{matrix} \Bigg) . \tag{18}$$

**Lemma 2.** *We have the following, where* $\mathrm{trace}^{\pm}$, $\det^{\pm}$ *respectively denote the trace, determinant of $S^{\pm}_{\mathbf{v}}$.*

$$\frac{\mathrm{trace}^+ - \mathrm{trace}^-}{2} = \frac{\kappa^r}{\sin\phi} + \kappa^t \sin\phi, \quad \frac{\det^+ - \det^-}{2} = \frac{a^*\kappa^r v - a\kappa^t - 2a^t\tau^t v}{v^3 \sin\phi} \tag{19}$$

The principal directions on $\gamma^{\pm}$ can be expressed in terms of the geometry of $\gamma$ by using Lemma 1.

## 7 The $A_1^3$ (Y-junction Curve) Case Continued

Now we obtain the complete set of constraints on the medial axis at a Y-junction point at the level of second order derivatives, using the results of Sect. 6. In order to equate $\gamma_i^+$, $\gamma_{i+1}^-$ up to second order derivatives we need to set $N_i^+ = N_{i+1}^-$ at Y-junction points (this gave Prop. 2) and, in addition, equate the principal curvatures and principal directions of $\gamma_i^+$, $\gamma_{i+1}^-$ at the Y-junction point $\gamma_i(r_0, 0)$.

Now we have three smooth sheets $\gamma_i$ for $i = 1$ to 3, so let $S^{\pm}_{\mathbf{v}_i}$ denote the matrix representation of the radial shape operator of $\gamma_i$ corresponding to boundary $\gamma_i^{\pm}$ with respect to the basis $\mathbf{v}_i = \{T_i, U_i\}$ at $\gamma_i(r_0, 0)$. Then let $\mathrm{trace}_i^{\pm}$, $\det_i^{\pm}$ be the trace, determinant of $S^{\pm}_{\mathbf{v}_i}$. Hence we give each term of (19) and (18) the subscript $i$. We can prove the following.

**Lemma 3.** *Equating principal curvatures on* $\gamma_i^+ = \gamma_{i+1}^-$ *at* $(r_0, 0)$ *is the same as setting* $\text{trace}_i^+ = \text{trace}_{i+1}^-$ *and* $\det_i^+ = \det_{i+1}^-$.

Using Lemmas 2 and 3 we get that equating principal curvatures on $\gamma_i^+ = \gamma_{i+1}^-$ at $(r_0, 0)$ implies Prop. 1.

In order to obtain all of the information given by equating principal curvatures and principal directions, it was necessary to use the second coordinate system of Sect. 3 (involving the $A_1^3$ curve). This allowed us to obtain the following (full details can be found in [17]).

**Proposition 5 (Constraint on Curvatures).** *At* $\gamma_i(r_0, 0)$ *on the Y-junction curve the principal curvatures and principal directions of the boundaries* $\gamma_i^+$, $\gamma_{i+1}^-$ *for* $i = 1, 2, 3$ *are equal if and only if the following holds.*

$$\begin{aligned}\frac{\kappa_i^W \sin\phi_Y}{\sin\theta_i} = {} & \frac{(\theta'_{i+1}\cos\phi_Y\cos 2\theta_{i+1} - \dot{\phi}_{i+1}\cos\phi_{i+1})}{\cos\theta_{i+1}\sin\theta_{i+1}} + \frac{2(\tau+\psi'_i)\cos\phi_Y\cos\theta_i}{\sin\theta_i} \\ & - \frac{(\theta'_{i+2}\cos\phi_Y\cos 2\theta_{i+2} - \dot{\phi}_{i+2}\cos\phi_{i+2})}{\cos\theta_{i+2}\sin\theta_{i+2}} \\ & + \frac{\kappa}{\sin\phi_Y}\left(\frac{\sin\psi_{i+2}\cos^2\phi_{i+2}}{\cos\theta_{i+2}} - \frac{\sin\psi_{i+1}\cos^2\phi_{i+1}}{\cos\theta_{i+1}} - \frac{\cos\psi_i B_i}{\sin\theta_i}\right)\end{aligned} \tag{20}$$

*where* $B_i = \cos^2\phi_Y + \sin^2\phi_Y\sin^2\theta_i$.

When the denominators of the expression above for $\kappa_i^W$ are zero, we need an alternative. A point for which $\sin\theta_i = 0$ for some $i$ is a fin $(A_1A_3)$ point and will be covered in Sect. 8. When $\sin\phi_Y = 0$ the three points of contact on the boundary are coincident and so the sphere of contact has $A_5$ contact with the boundary. This case is not generic for a surface, so we ignore it. The remaining possibility is when $\cos\theta_i = 0$ for some $i$. See [17] for an alternative form of (20) which does not have $\cos\theta_i$ in any of the denominators involved.

**Local Maximum or Minimum of $r$ Along a Y-junction Curve.** This corresponds to $\cos\phi_Y = 0$ at $\gamma_i(r_0, 0)$. Let $\cos\phi_Y = 0$ in (9) and we get

$$\sum_{i=1}^{3}\left(\frac{\kappa_i^W}{\sin\theta_i}\right) = 0\ . \tag{21}$$

Compare this with (1) in two dimensions.

## 8 The $A_1A_3$ (Fin) Case

An $A_1A_3$ point on the medial axis is where a Y-junction curve and an $A_3$ curve meet and end, i.e. the $A_1A_3$ point is the centre of a sphere which is tangent to a surface in three places, but two of the points of contact are coincident. The medial axis near to an $A_1A_3$ point looks like part of a swallowtail surface with a 'fin' (another medial sheet) intersecting with the swallowtail in a curve (see Fig. 2, right). The $A_1A_3$ point, or fin point, is then the point where the fin meets

the other two sheets. The equation (20) holds at Y-junction points, and so it holds in the limit as two of the points coincide.

Let the coincident points of contact be $\gamma_1^+$ and $\gamma_1^-$, which means that $\sin\phi_1 = 0$ at the $A_1A_3$ point, from (2) and (4). From (7) this corresponds to $\sin\theta_1 = 0$, (we dismissed the possibility of $\sin\phi_Y = 0$ just after Prop. 5). Therefore $\gamma_2$ and $\gamma_3$ are part of the swallowtail surface. Explicit calculations show that, as $t \to 0$, $\psi'_{2,3} \to \infty$ like $1/t$ and $\kappa^W_{2,3} \to \infty$ like $1/t^2$. Also $\psi'_2 \sin\theta_1 \to -\psi'_3 \sin\theta_1$ and $\kappa^W_2 \sin^2\theta_1 \to -\kappa^W_3 \sin^2\theta_1$, both of which are finite. The medial sheet $\gamma_1$ is smooth so $\kappa^W_1$, $a_1$, $a^t_1$, $a^*_1$ all remain finite as $\sin\theta_1 = 0$. Using these it can be shown that $\phi'_Y$, $\dot\phi_1 \sin\theta_1$, and $\psi'_1 \sin\theta_1$ all remain finite as $\sin\theta_1 \to 0$. Then we get the limiting forms of (20) for $i$ =1, 2, 3 as follows.

**Proposition 6.** *As we tend towards a fin point along a Y-junction curve, the medial sheets must satisfy the following, where each term is finite as* $\sin\theta_1 \to 0$.

$$\dot\phi_2 \sin^2\theta_1 \to \dot\phi_3 \sin^2\theta_1 \text{ and } (\dot\phi_2 - \dot\phi_3)\sin\theta_1 \to \text{ a finite number }, \tag{22}$$

$$\begin{aligned}\frac{\left(\kappa^W_2 \sin^2\theta_1\right)\sin\phi_Y}{\sin\theta_2} &\to -\frac{\left(\kappa^W_3 \sin^2\theta_1\right)\sin\phi_Y}{\sin\theta_2} \\ &\to -\frac{\dot\phi_2 \sin^2\theta_1 \cos\phi_2}{\cos\theta_2 \sin\theta_2} - 2\psi'_2 \sin\theta_1 \cos\phi_Y - \dot\phi_1 \sin\theta_1 \ ,\end{aligned} \tag{23}$$

$$\begin{aligned}\kappa^W_1 \sin\phi_Y \to\ & \frac{\cos\phi_2}{\cos\theta_2 \sin\theta_2}\left(\dot\phi_3 - \dot\phi_2\right)\sin\theta_1 - 2\left(\tau + \psi'_1\right)\cos\phi_Y \\ & + \frac{2\psi'_2 \sin\theta_1 \cos\phi_Y \cos(2\theta_2)}{\cos\theta_2 \sin\theta_2} - \frac{\kappa \cos\psi_1 \cos^2\phi_Y}{\sin\phi_Y} \ .\end{aligned} \tag{24}$$

See [17] for alternative forms of (23), (24) which do not have $\cos\theta_2$ in any of the denominators involved (and so are valid for $\cos\theta_i = 0$). The conclusions of Prop. 6 can be verified when the medial axis is as in the example of Sect. 3 (the fin point correponds to $y = 0$).

## 9 Summary and Conclusions

We have investigated the conditions which the geometry of the medial axis and the dynamics of the radius function must obey at singular points of the medial axis in order to obtain a consistent reconstruction of the boundary surfaces. The main case of interest is that of three sheets of the medial axis meeting in a curve (namely the locus of centres of spheres tangent to the boundary surface in three places). The constraints involve curvatures of the medial axis in two independent directions. To this extent they are reminiscent of the constraints in the 2D case which have been used in an important study of stochastic shape, among many others.

## References

1. H.Blum, 'Biological shape and visual science', *J. Theor. Biol.*, 38:205–287, 1973.
2. I.A.Bogaevsky, 'Perestroikas of shocks and singularities of minumum functions', *Physica D: Nonlinear Phenomena*, 173 (2002), 1-28.

3. S.Chen & R.E.Parent, 'Shape averaging and its applications to industrial design', *CGA*, 9(1):47–54, 1989.
4. J.Damon, 'Smoothness and geometry of boundaries associated to skeletal structures II: geometry in the Blum case', to appear in *Compositio Mathematica*.
5. J.Damon, 'Determining the geometry of boundaries of objects from medial data', Preprint, University of North Carolina at Chapel Hill, 2003.
6. P.J.Giblin & B.B.Kimia, 'On the intrinsic reconstruction of shape from its symmetries', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 (2003), 895-911.
7. P.J.Giblin and B.B.Kimia, 'A formal classification of 3D medial axis points and their local geometry', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (2004), 238-251.
8. U.Grenander & M.Miller, 'Computational anatomy: An emerging discipline', *Quarterly of Applied Mathematics*, LVI(4):617–694, December 1998.
9. L.Guibas, R.Holleman, & L. Kavraki, 'A probabilistic roadmap planner for flexible objects with a workspace medial axis based sampling approach', In *Proc. Intl. Conf. Intelligent Robots and Systems*, pages 254–260, Kyongju, Korea, 1999. IEEE/RSJ.
10. C.Holleman & L.Kavraki, 'A framework for using the workspace medial axis in PRM planners', In *Proceedings of the International Conference on Robotics and Automation*, pages 1408–1413, San Fransisco, CA, USA, 2000.
11. K.Leonard, PhD. thesis, Brown University, in preparation.
12. F.Leymarie, '3D Shape Representation via Shock Flows', PhD. thesis, Brown University, Providence, RI, USA. See http://www.lems.brown.edu/ leymarie/phd/
13. D.Mumford, 'The Shape of Objects in Two and Three Dimensions', *Gibbs Lecture 2003*, to appear in *Notices of the American Mathematical Society*.
14. P.L.Hallinan, G.G.Gordon, A.L.Yuille, P.J.Giblin, & D.Mumford, *Two- and Three-Dimensional Patterns of the Face*, A K Peters, Ltd. (1999).
15. M.Pelillo, K.Siddiqi, & S. Zucker, 'Matching hierarchical structures using association graphs', *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(11):1105–1120, November 1999.
16. S.M.Pizer & C.A.Burbeck, 'Object representation by cores: Identifying and representing primitive spatial regions', *Vision Research*, 35(13):1917–1930, 1995.
17. A.J.Pollitt, *Euclidean and Affine Symmetry Sets and Medial Axes*, Ph.D. thesis, University of Liverpool. In preparation.
18. T.B.Sebastian, P.N.Klein, & B.B.Kimia, 'Recognition of shapes by editing shock graphs', In *Proceedings of the Eighth International Conference on Computer Vision*, pages 755–762, Vancouver, Canada, July 9-12 2001. IEEE Computer Society Press.
19. K.Siddiqi, A.Shokoufandeh, S.Dickinson, & S.Zucker, 'Shock graphs and shape matching', *Intl. J. of Computer Vision*, 35(1):13–32, November 1999.
20. K.Siddiqi, K.J.Tresness, & B.B.Kimia, 'Parts of visual form: Ecological and psychophysical aspects', *Perception*, 25:399–424, 1996.
21. D.Siersma, 'Properties of conflict sets in the plane', *Geometry and Topology of Caustics*, ed. S. Janeczko and V.M.Zakalyukin, Banach Center Publications Vol. 50, Warsaw 1999, pp. 267–276.
22. J.Sotomayor, D.Siersma, & R.Garcia, 'Curvatures of conflict surfaces in Euclidean 3-space', *Geometry and Topology of Caustics*, ed. S. Janeczko & V.M.Zakalyukin, Banach Center Publications Vol. 50, Warsaw 1999, pp. 277–285.
23. S.C.Zhu and A.L.Yuille, 'FORMS: A flexible object recognition and modeling system', *Intl. J. of Computer Vision*, 20(3):187–212, 1996.

# Normalized Cross-Correlation for Spherical Images

Lorenzo Sorgi[1] and Kostas Daniilidis[2]

[1] Italian Aerospace Research Center
[2] University of Pennsylvania
{sorgi,kostas}@grasp.cis.upenn.edu

**Abstract.** Recent advances in vision systems have spawned a new generation of image modalities. Most of today's robot vehicles are equipped with omnidirectional sensors which facilitate navigation as well as immersive visualization. When an omnidirectional camera with a single viewpoint is calibrated, the original image can be warped to a spherical image. In this paper, we study the problem of template matching in spherical images. The natural transformation of a pattern on the sphere is a 3D rotation and template matching is the localization of a target in any orientation. Cross-correlation on the sphere is a function of 3D-rotation and it can be computed in a space-invariant way through a 3D inverse DFT of a linear combination of spherical harmonics. However, if we intend to normalize the cross-correlation, the computation of the local image variance is a space variant operation. In this paper, we present a new cross-correlation measure that correlates the image-pattern cross-correlation with the autocorrelation of the template with respect to orientation. Experimental results on artificial as well as real data show accurate localization performance with a variety of targets.

## 1 Introduction

Omnidirectional cameras [1,2] are now a commodity in the vision and robotics community available in several geometries, sizes, and prices and their images are being used for creating panoramas for visualization, surveillance, and navigation. In this paper we will address the problem of template matching in omnidirectional images. Our methods can be applied to any images which can be exactly or approximately mapped to a sphere. This means that our input images will be spherical images while the template can be given in any form which can be warped so that it has a support on the sphere. The main challenge in template matching is to be able to detect the template under as many geometric or illumination transformations as possible. This can be done by either comparing invariants like moment functions between the template and the image or with statistical techniques [3,4]. It is basic knowledge that we can compute affine invariants and that we can compute an affine transformation from combinations of image moments [5,6] or from Fourier descriptors [7]. Little work has been done on the computation of 3D-rotations from area-based features [8,9,10]. Even if template matching in planar image processing is certainly a problem studied in several different ways, there is hardly any work addressing this problem in omnidirectional imaging.

In this paper we consider 3D rotations, the natural transformations on the sphere, and we try to achieve a normalized cross-correlation to account for linear illumination

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 542–553, 2004.
© Springer-Verlag Berlin Heidelberg 2004

changes. A straightforward implementation of normalization in cross-correlation would be space variant, and as such quadratic in the number of image samples and cubic in the tessellation of the rotation space. The contribution of this paper is the development of a matcher that performs in the Fourier domain a sequence of two cross-correlations in order to guarantee the same performance as the normalized cross-correlation and simultaneously to avoid any space-variant operation. By introducing the concept of axial autocorrelation, we have been able to construct a new normalization that preserves the linearity of intensity transformations. The new measure consists of the image-pattern cross-correlation and its subsequent cross-correlation with the axial autocorrelation of the pattern.

The paper is split into 3 parts: Section 2 introduces the preliminary mathematics, Section 3 presents the matcher, and Section 4 analyzes results with real omnidirectional imagery.

## 2 Mathematical Preliminaries

This section collects the basic mathematical tools required for the formulation and solution of the spherical template matching problem. The proposed method in particular will make use of the analysis of real or complex valued functions defined on $S^2$, the two-dimensional unit sphere, and their harmonic representation. Throughout the paper we make use of the traditional spherical reference frame, according to which any point on $S^2$ is uniquely represented by the unit vector $\omega(\theta, \phi) = (\cos\phi \sin\theta, \sin\phi \sin\theta, \cos\theta)$. We denote with $\Lambda(R_{\alpha,\beta,\gamma})$ the linear operator associated with a rotation $R \in SO(3)$ and we use the ZYZ Euler angle parameterization for rotations.

The set of spherical harmonics $\{Y_m^l(\omega), l \geq 0, \mid m \mid \leq l\}$ (see [11]), forms a complete orthonormal basis over $L^2(S^2)$; thus any square-integrable function on the unit sphere $f(\omega) \in L^2(S^2)$ can be decomposed as series of spherical harmonics. The Spherical Fourier Transform and its inverse are defined [11] as:

$$f(\omega) = \sum_{l \in \mathbb{N}} \sum_{|m| \leq l} \hat{f}_m^l Y_m^l(\omega) \tag{1}$$

$$\hat{f}_m^l = \int_{\omega \in S^2} f(\omega) \overline{Y_m^l}(\omega) d\omega. \tag{2}$$

We assume for any spherical image to be square integrable on $S^2$ and bandlimited in order to use the sampling theorem for spherical function introduced by Driscoll and Healy [12]:

**Theorem 1 (Driscoll and Healy).** *Let $f(\omega)$ be a band-limited function, such that $\hat{f}_m^l = 0, \forall l \geq B$,, where B is the function bandwidth. Then for each $\mid m \mid \leq l < B$,*

$$\hat{f}_m^l = \frac{\sqrt{2\pi}}{2B} \sum_{j=0}^{2B-1} \sum_{k=0}^{2B-1} a_j^{(B)} f(\theta_j, \phi_k) \overline{Y_m^l(\theta_j, \phi_k)}$$

*where the function samples $f(\theta_j, \phi_k)$ are chosen from the equiangular grid: $\theta_j = j\pi/2B, \phi_k = k\pi/B$, and $a_j^{(B)}$ are suitable weights.*

Under a rotation $R(\alpha, \beta, \gamma)$, decomposed using the Euler angles parameterization, each harmonic of degree $l$ is transformed into a linear combination of only those $Y_m^l$ with the same degree $\mid m \mid \leq l$, [12]:

$$\Lambda(R_{\alpha,\beta,\gamma})Y_m^l(\omega) = \sum_{|k|\leq l} Y_k^l(\omega) D_{k,m}^l(R) \tag{3}$$

where $D_{m,k}^l(R)$ is defined as:

$$D_{m,k}^l(R) = e^{-im\alpha} d_{m,k}^l(\cos\beta) e^{-ik\gamma} \tag{4}$$

and for a definition of $d_{m,k}^l$, the irreducible unitary representation of SO(3), we refer to [13].

From (3) we can easily show that the effect of a rotation on a function $f(\omega) \in L^2(S^2)$ is a linear transformation associated with a semi-infinite block diagonal matrix in the Fourier domain:

$$h(\omega) = \Lambda(R_{\alpha,\beta,\gamma}) f(\omega) \Leftrightarrow \hat{h}_m^l = \sum_{|k|\leq l} \hat{f}_k^l D_{m,k}^l(R) \tag{5}$$

We finally add a lemma we first proved in [14] because it will be part of the first stage of our matcher.

**Lemma 1.** *Given $f(\omega), h(\omega) \in L^2(S^2)$, the correlation between $f(\omega)$ and $h(\omega)$ defined as*

$$C(\alpha, \beta, \gamma) = \int_{S^2} f(\omega) \Lambda(R_{\alpha,\beta,\gamma}) h(\omega) \; d\omega \tag{6}$$

*can be obtained from the spherical harmonics $\hat{f}_m^l$ and $\hat{h}_m^l$ via the 3-D Inverse Discrete Fourier Transform as*

$$C(\alpha, \beta, \gamma) = IDFT\{\sum_l \hat{f}_m^l \overline{\hat{h}_k^l} d_{m,h}^l(\pi/2) d_{h,k}^l(\pi/2)\}. \tag{7}$$

## 3 Spherical Pattern Matching

In this section we present normalized correlation and an approach on how to reduce the associated computational cost. We will introduce first the axial autocorrelation function for spherical functions and its invariant properties.

### 3.1 Autocorrelation Invariance

Let $f(\omega) \in L^2(S^2)$ be a function defined on the unit sphere. We define the axial autocorrelation function, as the autocorrelation of the function $f(\omega)$ computed rotating the function itself only around the Z axis, that points out of the North Pole $\eta$:

$$AC_{f,\eta}(\gamma') = \int_{S^2} f(\omega) \cdot \overline{\Lambda(R_{0,0,\gamma'}) f(\omega)} d\omega. \tag{8}$$

Expanding the function $f(\omega)$ as series of spherical harmonics we can rewrite relation (8) as:

$$AC_{f,\eta}(\gamma') = \int_{S^2} \sum_{l,m} \hat{f}_m^l Y_m^l \overline{\sum_{l',m',k} D_{m',k}^{l'}(0,0,\gamma') \hat{f}_k^{l'} Y_{m'}^{l'}} d\omega.$$

where $D_{m,k}^l(0,0,\gamma') = e^{-jk\gamma'}\delta_{m,k}$. Using the spherical harmonics orthogonality, we can finally simplify

$$AC_{f,\eta}(\gamma') = \sum_k \sum_l \left|\hat{f}_k^l\right|^2 e^{ik\gamma'} = IDFT\{\sum_l \left|\hat{f}_k^l\right|^2\}. \tag{9}$$

Under the action of a rotation $R_{\alpha,\beta,\gamma}$ the North Pole $\eta$ is transformed into the point $\omega' = (\beta, \alpha)$. Following the same procedure as above, we can show that the axial autocorrelation of the rotated function $h(\omega) = \Lambda(R_{\alpha,\beta,\gamma}) f(\omega)$, computed around the axis $\omega'$, is related to the function $AC_{f,\eta}$ by a simple translation:

$$AC_{h,\omega'}(\gamma') = AC_{f,\eta}(\gamma' - \gamma). \tag{10}$$

The proof of the previous identity is again straightforward if we rewrite the function $AC_{h,\omega'}(\gamma')$ using the expansion (1):

$$\begin{aligned} AC_{h,\omega'}(\gamma') &= \int_{S^2} \Lambda(R_{\alpha,\beta,\gamma}) f(\omega) \cdot \overline{\Lambda(R_{\alpha,\beta,\gamma'}) f(\omega)} d\omega = \\ &= \int_{S^2} \sum_{l,m,k} D_{m,k}^l(\alpha,\beta,\gamma) \hat{f}_k^l Y_m^l \cdot \overline{\sum_{l',m',k'} D_{m',k'}^{l'}(\alpha,\beta,\gamma') \hat{f}_{k'}^{l'} Y_{m'}^{l'}} d\omega = \\ &= \sum_{l,m,k,k'} d_{m,k}^l(\beta) d_{m,k'}^l(\beta) \hat{f}_k^l \overline{\hat{f}_{k'}^l} e^{-ik\gamma} e^{ik'\gamma'}. \end{aligned}$$

Using the orthogonality of the matrices $d^l(\beta)$:

$$\sum_m d_{m,k}^l(\beta) d_{m,k'}^l(\beta) = \delta_{k,k'}$$

we can finally simplify the $AC_{h,\omega'}(\gamma')$ expression:

$$AC_{h,\omega'}(\gamma') = \sum_k \sum_l \left|\hat{f}_k^l\right|^2 e^{ik(\gamma'-\gamma)} = AC_{f,\eta}(\gamma' - \gamma).$$

### 3.2 Normalized Cross-Correlation

Let $I(\omega)$ be a spherical image and $P(\omega)$ the template that is to be localized. If a pattern that matches exactly with the template is present in the image $I(\omega)$ at the position $\omega_0$, the linear filtering that maximizes the signal to noise ratio in $\omega = \omega_0$, can be expressed as a cross-correlation. Such a cross-correlation function $C : SO(3) \to \mathbb{R}$ for functions defined on the unit sphere (6) may be computed fast as a 3D Inverse Fourier transform of

a linear combination of the template and the image spherical harmonics (lemma 1). The difficulty is in normalizing this expression to account for variations of the local signal energy. A normalized cross-correlation coefficient would be written as

$$NC(R) = \frac{\int_{S^2}(I(\omega) - \bar{I}_w)(\Lambda(R_{\alpha,\beta,\gamma})P(\omega) - \bar{P})d\omega}{\sqrt{\int_{PW}\left|I(\omega) - \bar{I}_w\right|^2 d\omega \int_{PW}\left|P(\omega) - \bar{P}\right|^2 d\omega}}, \tag{11}$$

where PW is the image window defined by the support of the pattern, and the over bar means the mean value of the signals in the PW region. Hereafter, we use the overbar to represent the local mean of the image instead of the complex conjugate as we did in subsection 3.1.

Digital spherical images are processed after a projection to the uniformly sampled $\{\theta, \phi\}$ plane, where the distribution of samples is related to the chosen equiangular grid. With this representation the number of samples inside the overlapping window PW, and the shape of the window itself are space-variant. This means that to perform the normalization process shown in (11), we must refresh the shape of the window PW for every possible rotation which yields to an extremely high computational cost. In particular if $N_i^2$ is the number of image samples and $N_r^3$ is the dimension of the discretized rotational space $\{\alpha, \beta, \gamma\}$, then the computational cost of (11) is $O(N_i^2\, N_r^3)$. Observe that the cost is proportional to the number of sphere samples as well as the number of samples of $\alpha, \beta, \gamma$ because the entire sphere has to be revisited for every rotation due to the space variant pattern window.

A possible remedy to this problem has been introduced in [14] where the matching is performed between the image and template gradients:

$$C_G(R) = \int_{S^2} \langle \nabla_T[I(\omega)], \nabla_T[\Lambda(R_{\alpha,\beta,\gamma})P(\omega)]\rangle d\omega. \tag{12}$$

This approach, also based on space invariant operations, even if much faster than the normalized cross-correlation still does not guarantee robustness against any possible intensity linear transformation of the template intensity the same way a normalized cross-correlation would do. Taking advantage of the properties of the axial autocorrelation, we can instead modify the standard normalized cross-correlation matching and decrease the processing time without losing the normalization properties. Matching will be accomplished in two steps, the first of which is the image-template cross-correlation (6) followed by a 1D normalized cross-correlation performed using the axial autocorrelation as kernel. Let us suppose that a rotated version $P_r(\omega) = \Lambda(R_{\tilde{\alpha},\tilde{\beta},\tilde{\gamma}})P(\omega)$ of the template $P(\omega)$ is present in our image $I(\omega)$ (as shown Fig. 1.a), then for any value of $(\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})$ we can state that the following relation holds:

$$C(R)|_{\alpha=\tilde{\alpha},\beta=\tilde{\beta}} = C(\gamma) = AC_{P,\eta}(\gamma - \tilde{\gamma})$$

where $C(R)$ is the cross-correlation, computed with $\alpha = \tilde{\alpha}$ and $\beta = \tilde{\beta}$ and $AC_{P,\eta}(\gamma)$ is the template axial autocorrelation (Fig. 1.b).

The basic idea is then to perform the localization task using the normalized cross-correlation between the axial autocorrelation of the pattern and the image-pattern cross-correlation, instead of using the normalized correlation (11) between the pattern and the

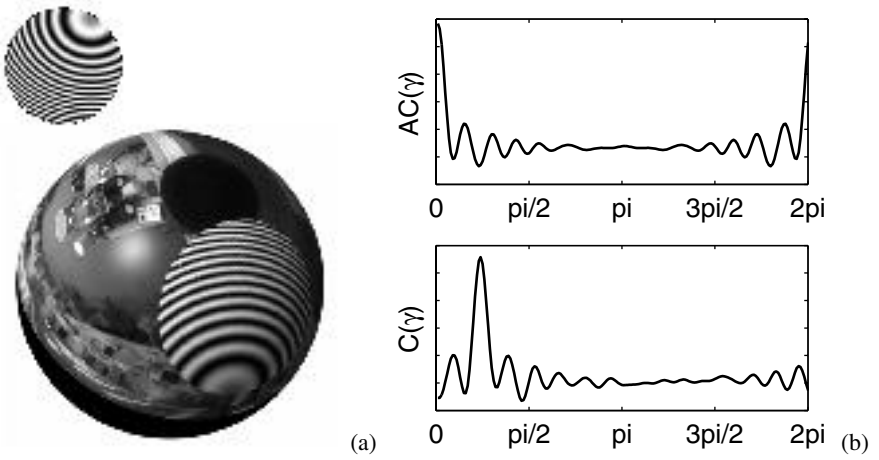



**Fig. 1.** On the left (a) a catadioptric image mapped on the sphere and an artificial pattern to be matched. (b) On the right the axial autocorrelation of the pattern $AC_{P,\eta}$ and the pattern-image cross-correlation $C(R)|_{\alpha=\tilde{\alpha},\beta=\tilde{\beta}}$, computed in the point $(\tilde{\alpha},\tilde{\beta})$ where the pattern is located.

image; this means that the matching is not performed on the pattern function but on its autocorrelation, (fig. 13):

$$M(\alpha,\beta) = \max_{\gamma' \in (0,2\pi]} \int_{\gamma} C(R_{\alpha,\beta,\gamma}) \cdot AC_{P,\eta}(\gamma - \gamma') d\gamma \tag{13}$$

We can show that if the pattern has undergone a linear intensity transformation then a linear transformation relates also their axial autocorrelations:

$$P'(\omega) = a \cdot P(\omega) + b \qquad \leftrightarrow \qquad AC_{P',\eta}(\gamma) = A \cdot AC_{P,\eta}(\gamma) + B$$

where $A = a^2$ and $B = 2ab \int_{S^2} P(\omega) d\omega + 4\pi b^2$. Besides since $AC_{P,\eta}$ and $C(R)|_{\alpha,\beta}$ are one-variable $(\gamma)$ discrete functions and the interval $\gamma \in [0, 2\pi)$ is uniformly sampled, the normalization process is thus reduced to a one-dimensional, space invariant operation.

We present now in detail a new algorithm for template matching on the sphere and the associated computational cost of each step. We assume that the rotational space can be sampled with $N_r^3$ samples and that the spherical image sampling $N_i^2$ is lower than $N_r^2$.

1. Initialize a $N_r \times N_r$ square localization map $M(\alpha,\beta)$. The dimension $N_r$ determines the Euler angle estimation precision, the rotational space will be in fact discretized in $N_r^3$ points.
2. Compute the spherical harmonics of both the image and the pattern. Using the algorithm described in [12] this operation require a computational cost $O(N_i \log^2 N_i)$, where $N_i^2$ is the number of samples of the uniformly sampled $(\theta,\phi)$-plane.

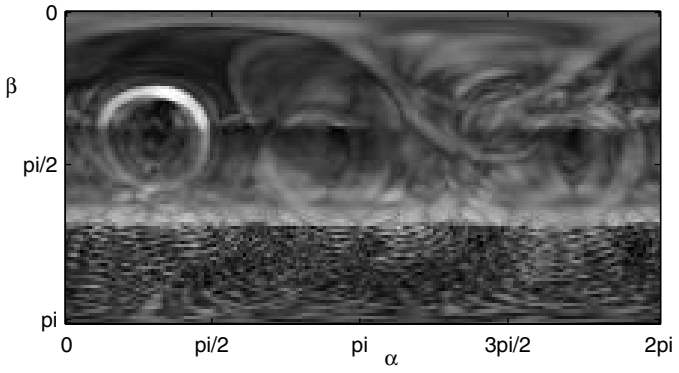



**Fig. 2.** The localization map $M(\alpha, \beta)$), relative to fig. 1.a.

3. Compute the axial autocorrelation of the template $AC_{P,\eta}$. This function can be obtained as 1D IDFT, so using the standard FFT requires only $O(N_r \log(N_r))$
4. Compute the cross-correlation $C(R)$ between the template and the image. Using Lemma 1 the computational cost of this step is $O(N_r^3 \log(N_r))$.
5. Build the localization map M: for every point $(\alpha, \beta)$ the normalized 1D circular cross-correlation between $AC_{P,\eta}$ and $C|_{\alpha,\beta}$ is computed, and the maximum value is assigned to the map. $M(\alpha, \beta) = max\{AC \otimes C|_{\alpha,\beta}\}$, (equation 13). Computing every normalized cross-correlation via 1D FFT the global computational cost of this step is again $O(N_r^3 log(N_r))$.
6. Localization of the pattern: exhaustive search of the max of the map $M(\alpha, \beta)$. Computational cost $O(N_r^2)$.
7. The estimation of the pattern orientation $\gamma$ is performed computing the shift factor between $AC_{P,\eta}$ and $C|_{\alpha,\beta}$. Computational cost $O(N_r)$.

The final computational cost is $O(N_r^3 \log(N_r))$ where $N_r^3$ is the sampling of the rotation space.

## 4 Experiments

In this section we will present some results of our algorithm applied to artificial as well as real images. Before pattern matching, an omnidirectional image is projected into the uniformly sampled $(\theta, \phi)$ space, required to compute the SFT using the sampling theorem (in any test has been chosen a square space of $700 \times 700$ pixels). The spherical Fourier transform has been performed assuming for the images to be band-limited, and we will use a subset of their spectrum in the third and fourth steps of the algorithm, considering a number of spherical harmonic up to the degree $l = 40$.

All the real pictures used to test the algorithm performance have been captured by a Nikon Coolpix 995 digital camera equipped with unique effective viewpoint catadioptric system (parabolic convex mirror and orthographic lens [1]). In [15], it is shown that

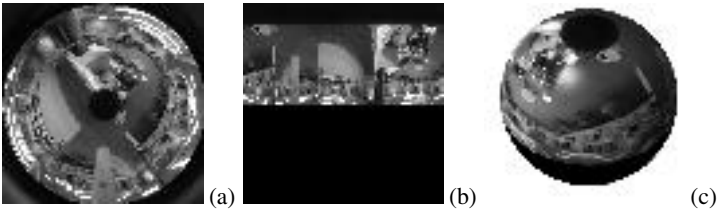
such a projection is equivalent to the projection on the sphere followed by a stereographic projection from the North Pole of the sphere to the equator plane. The coordinate transformation

$$u = \cot(\theta/2)\cos(\phi) \tag{14}$$

$$v = \cot(\theta/2)\sin(\phi) \tag{15}$$

allows us to map the catadioptric images directly on the unit sphere

$$I(\theta, \phi) = I(\cot(\theta/2)\cos(\phi), \cot(\theta/2)\sin(\phi)). \tag{16}$$

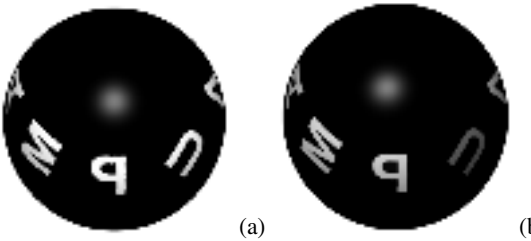


(a) (b) (c)

**Fig. 3.** On the left a 1400×1400 pixels catadioptric test-image. In the middle the same image presented in the equiangular $(\theta, \phi)$ plane and mapped on the sphere (on the right). The southern emisphere has been padded with zero-value samples.

**Table 1.** Euler angle estimation for the letters (Fig. 4). For every letter, it is indicated the real position (RP), the estimated position (EP1) using the original templates and the estimated position relative to the scaling-offset transformed version (Fig. 4.b), (EP2). In the last case we provide the indication of the different offset and scaling factors applied to each letter.

| | | A<br>sc.=0.6<br>offs.=25 | M<br>sc.=0.8<br>offs.=10 | P<br>sc.=0.9<br>offs.=-10 | U<br>sc.=0.5<br>offs.=-15 | V<br>sc.=0.7<br>offs.=20 |
|---|---|---|---|---|---|---|
| $\alpha$ | RP | $0^\circ$ | $45^\circ$ | $90^\circ$ | $135^\circ$ | $180^\circ$ |
| | EP1 | $0.89^\circ$ | $45.47^\circ$ | $91.82^\circ$ | $136.4^\circ$ | $180.98^\circ$ |
| | EP2 | $0.89^\circ$ | $45.75^\circ$ | $91.82^\circ$ | $136.4^\circ$ | $180.98^\circ$ |
| $\beta$ | RP | $60^\circ$ | $60^\circ$ | $60^\circ$ | $60^\circ$ | $60^\circ$ |
| | EP1 | $58.85^\circ$ | $58.85^\circ$ | $58.85^\circ$ | $58.85^\circ$ | $58.85^\circ$ |
| | EP2 | $58.85^\circ$ | $58.85^\circ$ | $58.85^\circ$ | $58.85^\circ$ | $58.85^\circ$ |
| $\gamma$ | RP | $0^\circ$ | $0^\circ$ | $270^\circ$ | $26^\circ$ | $120^\circ$ |
| | EP1 | $0.89^\circ$ | $0.89^\circ$ | $271.29^\circ$ | $25.98^\circ$ | $120.77^\circ$ |
| | EP2 | $0.89^\circ$ | $0.89^\circ$ | $271.29^\circ$ | $27.77^\circ$ | $120.77^\circ$ |



(a) (b)

**Fig. 4.** Two artificial images with five different letters. In (b) different scaling factors and offsets have been applied to the letters' intensity to test the matcher invariance to this class of transformation.

**Table 2.** Euler angle estimation for the real pattern (Fig.5). For every pattern is indicated the real position (RP) and the estimated position obtained with the proposed cascading normalized correlation (NC) compared with the analogous result obtained applying the gradient based correlation (GB). $NC_s$ and $GB_s$ refer to the matching performed using noisy images.

| | | Logo | Field | Paint | Park | Car | Door |
|---|---|---|---|---|---|---|---|
| $\alpha$ | RP | $92.57^\circ$ | $236.69^\circ$ | $77.18^\circ$ | $321.59^\circ$ | $257.27^\circ$ | $272.71^\circ$ |
| | NC | $93.61^\circ$ | $236.25^\circ$ | $77.56^\circ$ | $321.85^\circ$ | $257.66^\circ$ | $273.70^\circ$ |
| | GB | $95.39^\circ$ | $233.23^\circ$ | $81.13^\circ$ | $340.60^\circ$ | $257.66^\circ$ | $229.12^\circ$ |
| | $NC_n$ | $95.07^\circ$ | $237.04^\circ$ | $77.32^\circ$ | $320.7^\circ$ | $257.32^\circ$ | $270.0^\circ$ |
| | $GB_n$ | $315.63^\circ$ | $237.04^\circ$ | $143.24^\circ$ | $249.71^\circ$ | $152.66^\circ$ | $215.12^\circ$ |
| $\beta$ | RP | $64.28^\circ$ | $51.45^\circ$ | $51.45^\circ$ | $64.32^\circ$ | $66.89^\circ$ | $64.32^\circ$ |
| | NC | $62.40^\circ$ | $51.70^\circ$ | $51.70^\circ$ | $62.41^\circ$ | $65.97^\circ$ | $62.41^\circ$ |
| | GB | $68.30^\circ$ | $52.35^\circ$ | $71.32^\circ$ | $62.41^\circ$ | $67.88^\circ$ | $65.97^\circ$ |
| | $NC_n$ | $63.38^\circ$ | $48.17^\circ$ | $50.70^\circ$ | $63.38^\circ$ | $65.91^\circ$ | $63.38^\circ$ |
| | $GB_n$ | $78.59^\circ$ | $53.24^\circ$ | $70.98^\circ$ | $40.56^\circ$ | $42.16^\circ$ | $118.42^\circ$ |
| $\gamma$ | RP | $0^\circ$ | $26.37^\circ$ | $131.85^\circ$ | $80.25^\circ$ | $120.38^\circ$ | $28.66^\circ$ |
| | NC | $-0.89^\circ$ | $27.63^\circ$ | $133.50^\circ$ | $81.53^\circ$ | $120.77^\circ$ | $29.57^\circ$ |
| | GB | $-0.71^\circ$ | $28.45^\circ$ | $125.70^\circ$ | $95.60^\circ$ | $120.36^\circ$ | $38.54^\circ$ |
| | $NC_n$ | $1.28^\circ$ | $29.37^\circ$ | $134.04^\circ$ | $85.53^\circ$ | $121.28^\circ$ | $29.37^\circ$ |
| | $GB_n$ | $247.18^\circ$ | $206.62^\circ$ | $209.15^\circ$ | $97.60^\circ$ | $95.28^\circ$ | $246.12^\circ$ |

Naturally, the range of this mapping is limited by the field of view of the original catadioptric system: $\theta \in [0, 102^\circ), \phi \in [0, 360^\circ)$. The portion of the sphere that is not covered by the mapping has been padded with zero-value samples, (fig 3). We remark that the mapping of the catadioptric images onto the sphere is necessary to compute the spherical harmonics and perform the matching in the transformed space. Working directly with the original images in fact, would oblige us to deal with the negative effects of the distortion due to the template rotations. Such distortions of objects' shape in catadioptric images are extremely stressed and would certainly lead us to results of low quality and as shown above high processing time.

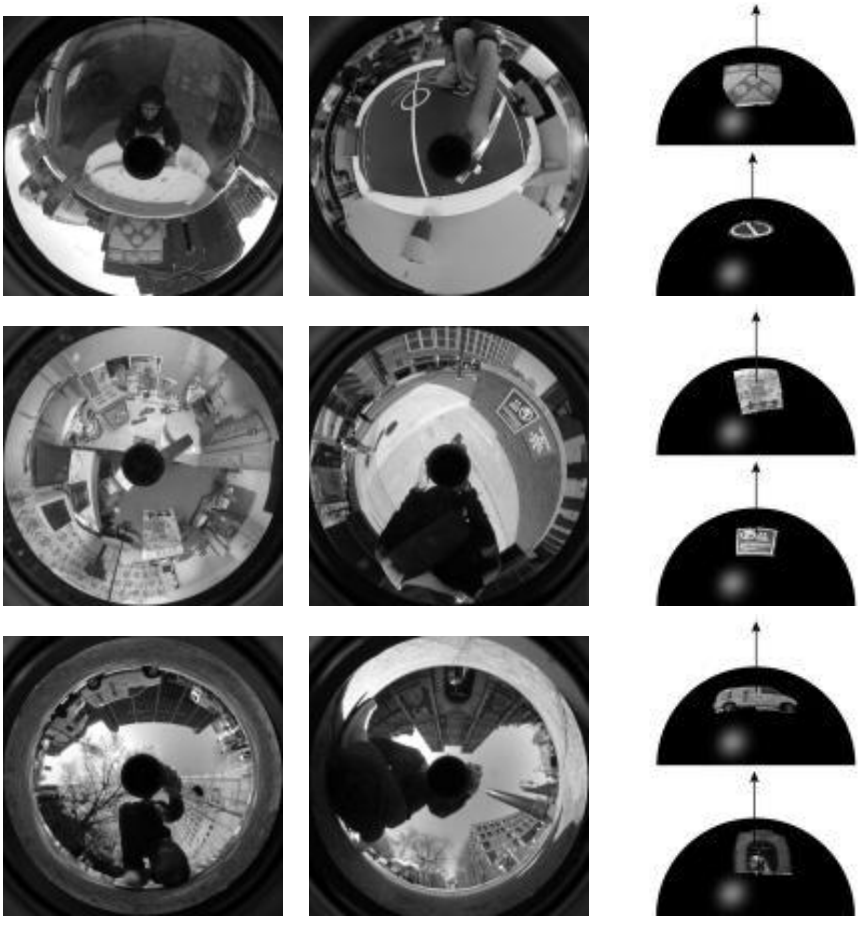


**Fig. 5.** Six catadioptric images. The patterns are presented mapped onto the sphere and slightly rotated forward to enhance their visibility.

We present first the localization performance of our normalized cross-correlation algorithm applied to an artificial image with five different letters A, M, P, U and V overlapped on a uniform background (Fig. 4.a). For every letter, in the second test, the grayscale value has been modified applying different scaling factors and offsets (Fig. 4.b). Table (1) presents real and estimated position of every letter. In this first test we do not present any comparison with other matching approaches because our main purpose is to show the effective scaling-offset invariance of the proposed method.

Then our algorithm has been tested using real images and real templates. As described in section 3.2, the matcher uses the pattern placed on the North Pole of the spherical reference frame as kernel to compute the cross-correlation function (6); however due to the structure of the catadioptric system used to take the pictures, it is impossible to

have a real image of the pattern placed on the North Pole. This problem has been overcome with an artificial rotation of the target pattern, selected from an omnidirectional image of the same environment captured with a different tripod orientation (rigid translational+rotational motion). Figure 5 shows six catadioptric images with the relatives selected objects and in the Table 2 we presents the localization performances. For comparison purposes, we performed the same test using the gradient matching introduced in [14], because it is the only matching system that requires the same computational cost of the algorithm presented in this paper and guarantees a good robustness to scaling-offset template transformation. The last test has been performed using a noisy version of the same real images of Fig. 5, obtained with an additive gaussian white noise (zero mean with 0.05 variance). Results are presented in the same Table 2. It is interesting to emphasize that using the axial autocorrelation kernel the template pose estimation is very robust to noisy images. This is due to the fact that the matching is performed by a harmonic analysis that involves a reduced number of transformed coefficients, which are less sensitive to noise.

## 5 Conclusion

In this paper we presented a pattern matching algorithm for spherical images based on the computation of a new cross-correlation measure: the cross-correlation of the image-pattern cross-correlation with the axial autocorrelation of the pattern. The new cross-correlation measure is invariant to linear intensity transformations and can be computed in $O(N_r^3 log(N_r))$ where $N_r^3$ is the sampling of the SO(3) Euler angle parameterization. This reduces the sampling cost significantly comparing to the cost associated with an explicit traversal of the image and the rotation space $O(N_i^2\, N_r^3)$. To validate the proposed algorithm we presented results obtained using artificial as well as real images and patterns, with the main goal to check the real maintenance of the matcher's invariance properties.

Just before the submission of the camera ready version of the paper we discovered a way to compute the local signal energy $\sqrt{\int_{PW} \left|I(\omega) - \bar{I}_w\right|^2 d\omega}$ required in the classical normalized cros-correlation coefficient (11) by computing the spherical harmonics of the support mask as well as the spherical harmonics of the square of the image and simultaneously keeping the same computational cost at $O(N_r^3 log(N_r))$. Time limits did not allow obtaining experimental results for this new procedure which will be thoroughly presented in a future report.

**Acknowledgements.** The authors are grateful for support through the following grants: NSF-IIS-0083209, NSF-IIS-0121293, NSF-EIA-0324977 and ARO/MURI DAAD19-02-1-0383.

## References

1. Nayar, S.: Catadioptric omnidirectional camera. In: IEEE Conf. Computer Vision and Pattern Recognition, Puerto Rico, June 17-19 (1997) 482–488

2. Baker, S., Nayar, S.: A theory of catadioptric image formation. In: Proc. Int. Conf. on Computer Vision, Bombay, India, Jan. 3-5 (1998) 35–42
3. Brunelli, R., Poggio, T.: Template matching: Matched spatial filters and beyond. Pattern Recognition **30** (1997) 751–768
4. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience **3** (1991) 72–86
5. Hall, E.: Computer Image Processing and Recognition. Academic Press (1979)
6. Sato, J., Cipolla, R.: Extracting group transformations from image moments. Computer Vision and Image Understanding **73** (1999) 29–42
7. Zahn, C., Roskies, R.: Fourier descriptors for plane close curves. IEEE Trans. Computers **21** (1972) 269–281
8. Lenz, R.: Rotation-invariant operators. In: Proc. Int. Conf. on Pattern Recognition, Paris, France, Sept. 28-3 (1986) 1130–1132
9. Tanaka, M.: On the representation of the projected motion group in 2+1d. Pattern Recognition Letters **14** (1993) 871–678
10. Segman, J., Rubinstein, J., Zeevi, Y.: The canonical coordinates method for pattern deformation: Theoretical and computational considerations. IEEE Trans. Pattern Analysis and Machine Intelligence **14** (1992) 1171–1183
11. Arfken, G., Weber, H.: Mathematical Methods for Physicists. Academic Press (1966)
12. Driscoll, J., Healy, D.: Computing fourier transforms and convolutions on the 2-sphere. Advances in Applied Mathematics **15** (1994) 202–250
13. Talman, J.D.: Special Funcitons. W.A.Benjamin Inc., Amsterdam (1968)
14. Sorgi, L., Daniilidis, K.: Template matching for spherical images. In: SPIE 16° Annual Symposium Electronic Imaging, San Jose, CA, January 18-22 (2004)
15. Geyer, C., Daniilidis, K.: Catadioptric projective geometry. International Journal of Computer Vision **43** (2001) 223–243

# Bias in the Localization of Curved Edges

Paulo R.S. Mendonça, Dirk Padfield, James Miller, and Matt Turek

GE Global Research
One Research Circle
Niskayuna, NY 12309
{mendonca,padfield,millerjv,turek}@research.ge.com

**Abstract.** This paper presents a theoretical and experimental analysis of the bias in the localization of edges detected from the zeros of the second derivative of the image in the direction of its gradient, such as the Canny edge detector. Its contributions over previous art are: a quantification of the localization bias as a function of the scale $\sigma$ of the smoothing filter and the radius of curvature $R$ of the edge, which unifies, without any approximation, previous results that independently studied the case of $R \gg \sigma$ or $\sigma \gg R$; the determination of an optimal scale at which edge curvature can be accurately recovered for circular objects; and a technique to compensate for the localization bias which can be easily incorporated into existing algorithms for edge detection. The theoretical results are validated by experiments with synthetic data, and the bias correction algorithm introduced here is reduced to practice on real images.

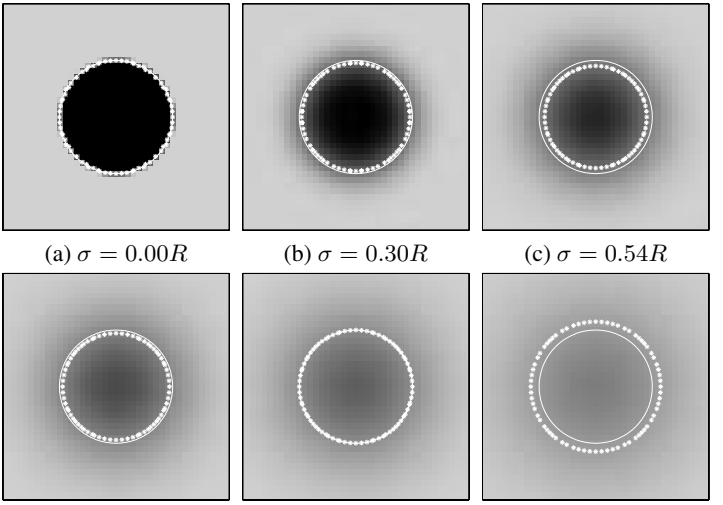
## 1 Introduction

Edge detection is a basic problem in early vision [16,21], and it is an essential tool to solve many high-level problems in computer vision, such as object recognition [25,20], stereo vision [17,1], image segmentation [13], and optical metrology [7].

In the application of edge detection to such high-level problems the criteria relevant to edge detector performance, as stated by Canny [5], are: 1) low error rate, meaning that image edges should not be missed and that false edges should not be found, and 2) good edge localization, meaning that the detected edge should be close to the true edge. The former criterion is dealt with in works such as [6,14,10]. It is this latter requirement that this research effort seeks to address. In particular, we quantify the error in edge localization as a function of the edge curvature. This error occurs even for noise-free images, and therefore the effect of noise was not considered in this paper.

With the exception of non-linear methods such as anisotropic diffusion [19], edge detection can almost invariably be reduced to convolution with a Gaussian kernel with given scale parameter $\sigma$ and computation of image gradients. For an ideal one-dimensional step [12] or for a two-dimensional step with rectilinear boundary [15], the location of a noise-free isolated edge can be exactly determined, and bounds can be provided for the error in the edge location in the presence of noise. However, when these models are applied to the analysis of the detection of *curved* edges in two-dimensions, effects due to the interaction between the kernel and the curved edges are not accurately captured. This problem has been tackled in previous works for particular intervals of $\sigma$ and the

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 554–565, 2004.
© Springer-Verlag Berlin Heidelberg 2004



(a) $\sigma = 0.00R$ (b) $\sigma = 0.30R$ (c) $\sigma = 0.54R$

(d) $\sigma = 0.70R$ (e) $\sigma = 0.80R$ (f) $\sigma = 1.00R$

**Fig. 1.** Canny edges, points of which are shown as white asterisks, detected on the images of a circle with constant radius $R = 10$ smoothed by Gaussian kernels of different standard deviations $\sigma$. The true location of the edge is shown as a solid curve. For $\sigma = 0.00R$ (a), the points along the detected edge coincide with the solid curve. It can be seen that the detected edge moves towards the center of curvature of the true edge as $\sigma$ increases (b), until it reaches a critical value $\sigma = \sigma_c = 0.54R$ (c) (see section 3.2). It then begins to move away from the center of curvature (d), and intercepts the true edge when $\sigma = \sigma_0 = 0.80R$ (e) (see, again, section 3.2). The detected edge continues to move outwards as $\sigma$ grows (f).

edge curvature [2,22,23], but the theoretical and experimental characterization of these effects for the full range of values of the scale $\sigma$ is first introduced in this paper. The results presented here show that the extent to which detected edges move from their ideal positions can be determined and reduced based on the scale parameter of the smoothing kernel and the radius of curvature of the detected edge. The prescription of an optimal scale for accurate estimation of edge location is also introduced.

Section 2 presents the mathematical results which are used throughout the remainder of the paper. An analysis of the bias in the localization of edges as a function of the scale of the smoothing kernel and the curvature of the edge is shown in section 3, which also brings an experimental validation of the theoretical derivations and analyzes effects that occur at certain critical values for the scale parameter $\sigma$. Section 4 introduces a method for correcting the localization bias. Experimental results are given in section 5, followed by a conclusion and proposals for future work in section 6.

## 2 Theoretical Background

The simplest 2D model that can be used to analyze errors in the localization of curved edges is a circle, since its border has constant curvature. The limit case of zero curvature

corresponds to the analysis of step edges, which can be carried out either implicitly, as in [5], which considered 1D step edges, or explicitly, as in [10]. The main steps common to most algorithms for edge detection, such as the ones cited in section 1, are image smoothing by convolution with an appropriate kernel, usually Gaussian, the computation of derivatives of the image, and the localization of peaks of the first derivative or, equivalently, zeros of the second derivative. Convolution and differentiation commute, and their combined effect is that of convolving the image with derivatives of the Gaussian kernel. The operation of convolution in space is equivalent to multiplication in the frequency domain, and therefore a brief review of some results regarding the Fourier analysis of circularly symmetric signals will be presented.

First, recall that the Fourier transform of a circularly symmetric function is also circularly symmetric, i.e., if $f(x,y) = f(\sqrt{x^2+y^2}) \leftrightarrow F(w_1,w_2)$ are a Fourier pair, then $F(w_1,w_2) = F(\sqrt{w_1^2+w_2^2})$. Moreover, if $r = \sqrt{x^2+y^2}$ and $\rho = \sqrt{w_1^2+w_2^2}$, then

$$F(\rho) = 2\pi \int_0^\infty r f(r) J_0(\rho r) dr \quad \text{and} \tag{1}$$

$$f(r) = \frac{1}{2\pi} \int_0^\infty \rho F(\rho) J_0(r\rho) d\rho, \tag{2}$$

where $J_n(\cdot)$ is the nth-order Bessel function of the first kind.

Consider the function $s : \mathbb{R}^2 \to \mathbb{R}$ given by

$$(x,y) \mapsto s(x,y) = \begin{cases} 1 & \text{if } x^2+y^2 \le R^2, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

This function is a circularly symmetric pulse, and its representation in polar coordinates, $s : \mathbb{R}^+ \times [0, 2\pi) \to \mathbb{R}$, is given by

$$(r,\theta) \mapsto s(r,\theta) = s(r) = \begin{cases} 1 & \text{if } r \le R, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Using (1), the Fourier transform of $s(r)$, denoted by $S(\rho)$, can be found as

$$S(\rho) = \frac{2\pi R}{\rho} J_1(R\rho). \tag{5}$$

Now let $k(x,y)$ be a Gaussian kernel, i.e., $k(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$. The function $k(x,y)$ is circularly symmetric, i.e., $k(x,y) = k(r)$, and therefore its Fourier transform can be computed from (1), resulting in $K(\rho)$ given by

$$K(\rho) = e^{\frac{-\sigma^2\rho^2}{2}}. \tag{6}$$

The product $\hat{S}(\rho)$ of (5) and (6) is

$$\hat{S}(\rho) = \frac{2\pi R}{\rho} e^{\frac{-\sigma^2\rho^2}{2}} J_1(R\rho), \tag{7}$$

and, therefore, using (2), the result $\hat{s}(r)$ of smoothing the function $s(r)$ with the Gaussian kernel $k(r)$ will be given by

$$\hat{s}(r) = R \int_0^\infty e^{\frac{-\sigma^2\rho^2}{2}} J_0(r\rho) J_1(R\rho) d\rho. \tag{8}$$

Although (8) cannot be computed in closed form, its derivatives can [24]. This observation was central for the development of this work. The first and second derivatives of (8) are shown below:

$$\frac{d\hat{s}}{dr} = -\frac{R}{\sigma^2} e^{\frac{-(r^2+R^2)}{2\sigma^2}} I_1\left(\frac{Rr}{\sigma^2}\right) \text{ and} \tag{9}$$

$$\frac{d^2\hat{s}}{dr^2} = -\frac{R}{\sigma^2} e^{\frac{-(r^2+R^2)}{2\sigma^2}} \left[\left(\frac{r}{\sigma^2} + \frac{1}{r}\right) I_1\left(\frac{Rr}{\sigma^2}\right) - \frac{R}{\sigma^2} I_0\left(\frac{Rr}{\sigma^2}\right)\right], \tag{10}$$

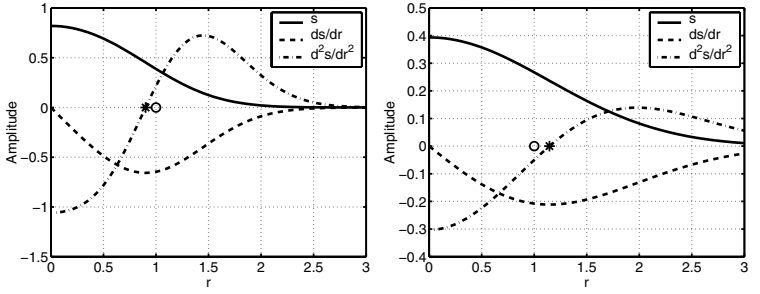
where $I_n(\cdot)$ is the modified Bessel function of order $n$. The problem of edge detection can be stated now as the search for the zeros of (10).

## 3 Bias in Edge Localization

Fig. 1 shows the experimental results of running a Canny edge detector on the image of a circularly symmetric pulse of radius $R = 10$ at scales $\sigma = 0.00R$, $0.30R$, $0.54R$, $0.70R$, $0.80R$, and $1.00R$. These results were obtained by generating a synthetic image of a disc. Partial volume effects were taken into account by subsampling pixels near the edge of the circle and counting the number of inside and outside samples. For each value of the scale $\sigma$ of the Gaussian smoothing filter a Canny edge detector with subpixel accuracy was run on the images. The mean distance from these points to the center of the circle was then computed and declared to be the detected radius.

It can be seen that, even in the absence of noise, the detection of edges by computing the zeros of the second derivative of an image in the direction of its gradient produces a shift in the localization of the edges, which is a well-known effect [21]. By repeating this experiment at various realistic levels of resolution and quantization, it was found that these parameters have minimal influence on localization accuracy. The results in [22,23] show that, if $R \gg \sigma$, the shift in the location of edges will be towards their center of curvature. In [8,9], a similar result was found by using a pair of straight edges with a common endpoint and varying intersection angle as the model for the analysis. In the case were $R \ll \sigma$, the analysis in [2] shows that the shift will be in the opposite direction, the so called "expansion phenomenon" discussed in section 3.2. These works did either purely numerical or approximate analytical studies and proposed independent remedies to the problem. The analysis in section 3.2 will unify these results, producing an exact result for the shift in edge location that is *valid for any value of* $R/\sigma$.

An important observation is that if $r_0$ is a zero of (10), the value of $r_0/\sigma$ depends only on the ratio $R/\sigma$, i.e., if the radius of the pulse and the scale of the smoothing kernel are multiplied by a common factor, the value of $r_0$ will be multiplied by the same factor.




(a) $\sigma = 0.54R$ (b) $\sigma = 1.00R$

**Fig. 2.** Profiles of a smothed circular pulse and its derivatives along the radial direction. The amplitude of the pulse is 1 and its radius $R$ is 1. The profiles are shown as a function of $r$, as in (8). The circle marks the ideal location for the edge of the pulse, and the asterisk indicates the location of an edge corresponding to the zero crossing of the second derivative of the pulse. Comparing the position of the zero-crossings with the ideal locations of the edges, it can be seen that when $\sigma = 0.54R$ the radius of the edge is underestimated by 9.6%, whereas for $\sigma = 1.00R$ the radius is overestimated by 14.5%. Observe that the full 2D pulse and its derivatives in the direction of its gradient can be obtained by rotating the respective curves around the vertical axis of the plots. The experimental results shown in Fig. 1 produced, for $\sigma = 0.54R$, an underestimation of the radius of curvature of 9.7%, and for $\sigma = 1.00R$ an overestimation of 14.7%. The small deviation from the theoretical predictions are attributable to sampling and quantization.

### 3.1 Theoretical versus Experimental Bias

Fig. 2 shows a plot of (8) for a pulse with radius $R = 1.0$ smoothed by a Gaussian kernel with scale $\sigma = 0.54R$ in (a) and $\sigma = 1.00R$ in (b). The images also show the first and second derivatives of the pulse as functions of $r$, as given by (9) and (10). The zero-crossings are, in both cases, indicated by asterisks, and the true positions of the edges are marked by a circle. The location of the detected edge underestimates the true radius by 9.6% when $\sigma = 0.54R$, and overestimates it by 14.5% when $\sigma = 1.00R$. These results can be compared with the experimental errors obtained for the detected radius in Fig. 1, which were 9.7% of underestimation when $\sigma = 0.54R$ and 14.7% of overestimation when $\sigma = 1.00R$. The small deviation is due to effects not considered in the theoretical model, such as sampling and quantization.

It is clear that there is a variable offset in the location of edges obtained by an edge detector based on zero crossings of the second derivative in the gradient direction. Contrary to what is commonly believed, this offset can produce either under- or overestimation of the edge's radius of curvature, unifying the results in [2,22,23].

### 3.2 Critical Values of $\sigma$

As $\sigma$ varies in the interval $[0, \infty)$, which is equivalent to a decrease in the ratio $R/\sigma$ from $\infty$ to 0 for a fixed $R$, it can be seen from Fig. 1 that, initially, the detected edge moves towards the center of curvature of the original edge. As $\sigma$ increases to a certain

**Table 1.** Critical values of $\sigma$. As $\sigma$ varies from 0 to $0.543R$, the detected edge shifts towards the center of curvature of the true edge, and its radius of curvature $r_0$ varies from $R$ to $0.901R$. When $\sigma$ reaches the critical value of $\sigma_c = 0.543R$, the detected edge begins to move outwards, until it crosses the position of the original edge at $\sigma_0 = 0.804R$. As $\sigma$ keeps increasing, the detected edge still moves outwards, until, as $\sigma \to \infty$, the radius $r_0 \to \sigma$.

| Range of $\sigma$ | Corresponding $r_0$ | Direction of motion | |
|---|---|---|---|
| $0 \leq \sigma < 0.543R = \sigma_c$ | $R \geq r_0 > 0.901R$ | $\partial r_0/\partial\sigma < 0$ | towards center |
| $\sigma = \sigma_c = 0.543R$ | $r_0 = 0.901R$ | $\partial r_0/\partial\sigma = 0$ | no motion |
| $\sigma_c = 0.543R < \sigma < 0.804R = \sigma_0$ | $0.901R < r_0 < R$ | $\partial r_0/\partial\sigma > 0$ | away from center |
| $\sigma = \sigma_0 = 0.804R$ | $r_0 = R$ | $\partial r_0/\partial\sigma > 0$ | away from center |
| $\sigma_0 = 0.804R < \sigma < \infty$ | $R < r_0$ | $\partial r_0/\partial\sigma > 0$ | away from center |
| $\sigma \to \infty$ | $r_0 \to \sigma$ | $\partial r_0/\partial\sigma > 0$ | away from center |

$\sigma = \sigma_c$, the edge reaches a critical point after which it starts moving back towards the location of the original edge. At a particular $\sigma = \sigma_0 > \sigma_c$, the location of the detected edge coincides with that of the original edge, and as $\sigma$ keeps increasing, the detected edge continues to shift outwards. The values of $R/\sigma$ at which these events occur can be directly computed from (9) and (10), as will be shown next.

**Optimal Scale for Curvature Resolution.** The zeros of (10) give the location of the edges of (8). Observe that the function $h(r; R, \sigma)$ given by

$$h(r; R, \sigma) = \left(\frac{r}{\sigma^2} + \frac{1}{r}\right) I_1\left(\frac{Rr}{\sigma^2}\right) - \frac{R}{\sigma^2} I_0\left(\frac{Rr}{\sigma^2}\right) \tag{11}$$

is the only term that influences the location of the zeros of (10), since the factor $-R/\sigma^2$ is constant and the factor $\exp(-(r^2 + R^2)/(2\sigma^2))$ is positive for all $r$. The non-zero value of $\sigma$ that results in minimal bias in edge localization, denoted $\sigma_0$, can be found by making $r = R$ in (11) and solving

$$\left(\frac{R}{\sigma^2} + \frac{1}{R}\right) I_1\left(\frac{R^2}{\sigma^2}\right) - \frac{R}{\sigma^2} I_0\left(\frac{R^2}{\sigma^2}\right) = 0 \tag{12}$$

for $\sigma$. Using the substitution $R/\sigma = \alpha$, (12) can be rewritten as

$$\alpha^2 I_0(\alpha^2) - (1 + \alpha^2) I_1(\alpha^2) = 0, \tag{13}$$

with solution $\alpha \approx 1.243$, i.e., $\sigma_0 \approx 0.804R$. The value $\sigma_0$ is of great importance. The localization of edges with radius $R = 1.243\sigma_0 = \sigma_0/0.804$ for any $\sigma_0$, although still affected by noise, sampling and quantization, *will not be subject to the effect of any curvature-induced bias*, according to the theoretical model. This result complements that of [10], which provides a minimum scale $\hat{\sigma}$ at which image gradients can be reliably detected. In that work, however, edges are modeled as steps with rectilinear boundaries, which lead the authors to conclude that [10, section 9] "localization precision improves monotonically as the maximum second derivative filter scale is increased." This statement is adjusted by the results of this work, which relates localization precision to the curvature of the detected edge.

**Maximum Shift to the Center of Curvature.** To obtain $\sigma_c$ it is necessary to find the simultaneous solution of (11) and $\frac{\partial r}{\partial \sigma} = 0$, i.e., the scale for which the *drift velocity* [14] of the edge with respect to scale is zero. This condition can be rewritten [11, section 5.6.2] after the appropriate simplification as

$$\left(\frac{R}{r} + \frac{\sigma^4}{r^3 R}\right) I_1\left(\frac{Rr}{\sigma^2}\right) - I_0\left(\frac{Rr}{\sigma^2}\right) = 0. \tag{14}$$

By making the substitution $R/\sigma = \alpha$ and $r/\sigma = \beta$, the simultaneous set of equations given by (11) and (14) becomes

$$\left(\frac{\beta}{\alpha} + \frac{1}{\alpha\beta}\right) I_1(\alpha\beta) - I_0(\alpha\beta) = 0, \tag{15}$$

$$\left(\frac{\alpha}{\beta} + \frac{1}{\alpha\beta^2}\right) I_1(\alpha\beta) - I_0(\alpha\beta) = 0, \tag{16}$$

with solution $\alpha \approx 1.842$ and $\beta \approx 1.660$. Therefore, the minimum value of $r_0$ is $\approx 0.901R$, obtained when $\sigma = \sigma_c \approx 0.543R$. The significance of the value $\sigma_c$ and the corresponding radius is that they provide a limit to the maximum bias in the direction of the center of curvature, or a *one-sided bound to the bias in the localization of an edge as a function of its radius of curvature*.

**Expansion Phenomenon.** The expansion phenomenon described in [2] can also be analyzed with the techniques developed here. Assume that $\sigma \to \infty$, i.e., $\alpha \to 0$, with $\alpha$ and $\beta$ as in the previous paragraph. The first order Taylor expansion of $I_1(x)$ and $I_0(x)$ are $I_1(x) = x/2$ and $I_0(x) = 1$. Substituting these expressions in (15), one obtains $\left(\frac{\beta}{\alpha} + \frac{1}{\alpha\beta}\right)\frac{\alpha\beta}{2} - 1 = \frac{\beta^2+1}{2} - 1 = 0$, producing the result $\lim_{\alpha\to 0} \beta = 1 \Rightarrow r_0 = \sigma$. This shows that, as $\sigma$ increases, *closed contours will tend to become circles with radius* $\sigma$, as described in [2]. Table 1 summarizes the critical values of $\sigma$ and the corresponding values of $r_0$.

## 4 Correction of Bias in Edge Localization

Because the value of $r_0/\sigma$ depends only on the ratio $R/\sigma$, it is possible to summarize the relationship between the inputs $R$ and $\sigma$ and the output $r_0$ in a single one-dimensional plot of $(r_0 - R)/\sigma$ as a function of $r_0/\sigma$, shown in Fig. 3. The parameter $r_0/\sigma$ was chosen as the abscissa for convenience: the curves in Fig. 3 can then be interpreted as a lookup table to correct the position of a point in an edge as a function of the scale $\sigma$ of the smoothing kernel and the measured radius of curvature of the detected edge at the location of the given point. The dashed and solid curves in the figure represent the theoretical and experimental lookup tables, respectively.

This suggests a simple algorithm to correct for curvature-induced bias in the localization of edges. First, edge detection by locating the zero crossings of the second derivative of the image in the direction of its gradient is performed, at a scale $\sigma_e$. Then, the *blur parameter* $\sigma_b$ of the image, sometimes referred to as the point spread function,

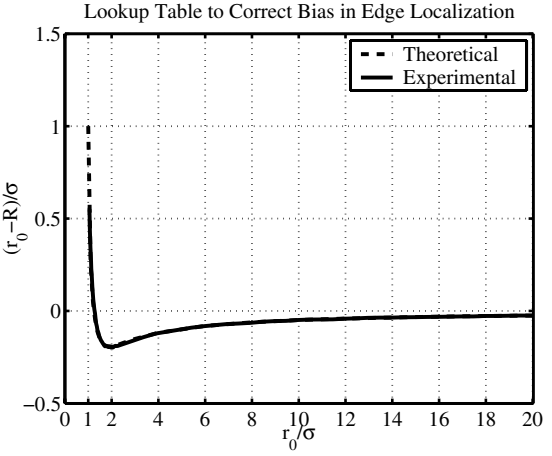



**Fig. 3.** Theoretical (dashed line) and experimental (solid line) look up tables to correct the location of edges produced by an edge detector based on finding the zeros of the second derivative of the image in the gradient direction. The theoretical curve was produced by numerically solving (11) for the detected radius $r_0$, for different values of the ratio $R/\sigma$. The experimental curve was obtained by applying a Canny edge detector to an image such as the one shown in Fig. 1(a). The use of this look up table to correct the bias in the localization of edges is described in Alg. 1. The theoretical and experimental curves overlap almost completely.

is estimated by using a method such as the one in [18]. The combination of $\sigma_{\mathrm{e}}$ and $\sigma_{\mathrm{b}}$ produces an effective scale $\sigma$ given by $\sigma = \sqrt{\sigma_{\mathrm{e}}^2 + \sigma_{\mathrm{b}}^2}$, which represents an estimate of the total smoothing of the image. The radius of curvature $r_0$ of the detected edges is then estimated for each point along the edges by fitting a circle to the point and a number of its closest neighbors. Besides determining the radius of curvature, this procedure also locates the center of curvature, which corresponds to the center of the fitted circle. For each point along the edges, the ratio $r_0/\sigma$ is computed, and the bias at that point is estimated from the lookup table in Fig. 3. The point is then shifted along the line that connects it to its center of curvature, according to the output of the lookup table. This procedure, summarized in Alg. 1, is repeated for all points in the detected edges.

## 5 Experimental Results

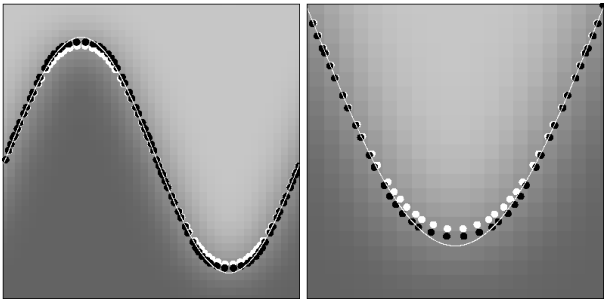
In order to validate the technique introduced in this paper it is necessary to demonstrate its usefulness for edges of generic shape. The development of the theoretical model assumes circularly symmetric images, and, to put the algorithm to test, an experiment was run on an image with a sinusoidal edge, shown in Fig. 4. The radius of curvature of the ideal sinusoid that separates the light and dark regions spans the interval $[2.5, \infty)$ in pixel units, with the minimum radius being reached at peaks of the edge, and the maximum radius obtained at inflection points. A Canny edge detector was run on the image, and each point along the detected edge was shifted appropriately, according to

**Algorithm 1** Correction of bias in edge localization.

1: detect edges as zero-crossings of second derivative in gradient direction with scale $\sigma_{\mathrm{e}}$;
2: estimate the image blurring factor $\sigma_{\mathrm{b}}$ and the effective scale $\sigma = \sqrt{\sigma_{\mathrm{e}}^2 + \sigma_{\mathrm{b}}^2}$;
3: **for** each point $\mathbf{x}$ in detected edges **do**
4: compute radius of curvature $r_0$ of detected edge at $\mathbf{x}$;
5: compute center of curvature $\mathbf{x}_0$ of detected edge at $\mathbf{x}$;
6: compute $r_0/\sigma$;
7: determine bias $\epsilon = (r_0 - R)/\sigma$ from the lookup table in Fig. 3;
8: correct $\mathbf{x}$ by moving it $-\epsilon\sigma$ pixel units in the direction $(\mathbf{x} - \mathbf{x}_0)/\|\mathbf{x} - \mathbf{x}_0\|$;
9: **end for**

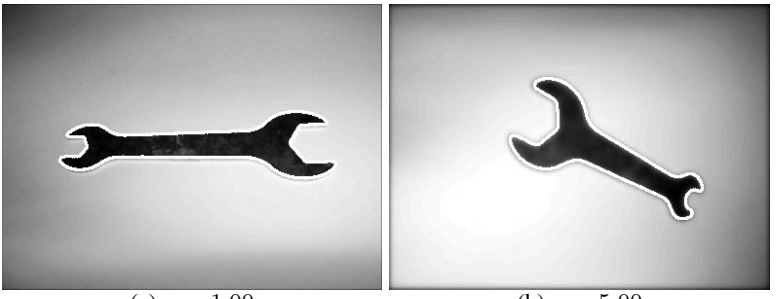
Alg. 1. The detected edge for $\sigma = 3$ is shown as a series of white dots in Fig. 4, and the corresponding corrected points are shown in black. It can be seen that, although still non-zero, the bias has been significantly reduced. To quantify this improvement, the RMS distance between the detected points and the ideal edge was computed before and after correction. For $\sigma = 1, 2, 3$ and $4$, the original RMS distances were 0.1266, 0.2828, 0.5067 and 0.8097, respectively. After correction, the RMS distances were reduced to 0.1024, 0.1087, 0.2517 and 0.7163. The error is not zero because, even though the curve can be locally approximated by circular patches, the edge is not circularly symmetric as assumed in the model. It is worth noting that, for the commonly used $\sigma$ values of 1 and 2, the anecdotal standard of $1/10$ of a pixel unit in accuracy is achieved *only after* the application of the technique for bias correction introduced in this paper.

An experiment in invariant theory was conducted to evaluate the performance of the algorithm discussed here when applied to real data. Two images of a wrench were acquired at different viewpoints. A Canny edge detector was used to detect edges on the images, each image with a different kernel scale $\sigma = 1.00$ and $\sigma = 5.00$. The smoothed images and the overlaid edges are shown in Fig. 5. Using the technique developed in



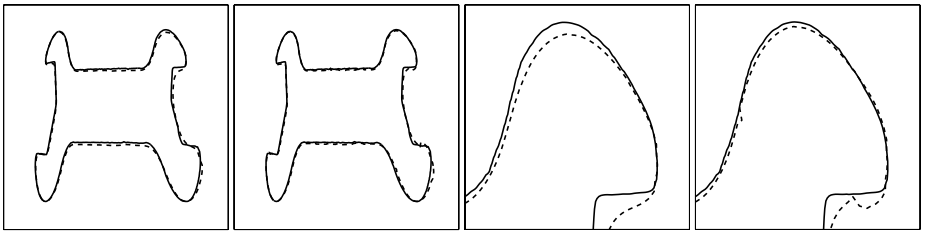
(a) (b)

**Fig. 4.** Detection of sinusoidal edges for $\sigma = 3$. The radius of curvature of the ideal edge separating the light from dark areas, shown as a solid white line, varies from 2.5 to $\infty$ in pixel units. The detected and corrected edges are indicated by white and black dots, respectively. Notice the effect of the correction in the zoomed image in (b).



(a) $\sigma = 1.00$ (b) $\sigma = 5.00$

**Fig. 5.** Two viewpoints of a wrench used to demonstrate the improvement in accuracy in the detection of edges. Images (a) and (b) were smoothed with scales of 1 and 5, respectively. The detected edge is shown as a thick white line on the border of the wrench. The corrected edges were not shown because, at this resolution, the difference between the detected and corrected edges is not visible.



(a) (b) (c) (d)

**Fig. 6.** Comparison between the projectively invariant signature of the image contour in Fig. 5(a) (solid line) versus that of Fig. 5(b) (dashed line). Image (a) shows the signatures of the detected edges and image (b) shows the signatures of the corrected edges. Notice that the signatures in image (b) are closer together, indicating that the corrected points are closer to the ideal edge of the wrench. Images (c) and (d) show zoomed in portions of the detected and corrected signatures, respectively. Notice the improvement in image (d) over image (c).

[25], the projectively invariant signatures of the edges of the wrench in each view were computed, as can be seen in Fig. 6(a). Due to the different scales used in the detection of the edges, their invariant signatures are significantly different. This difference is visibly reduced if the edges, before the computation of their respective signatures, are corrected with the algorithm introduced in this paper, as can be seen in Fig. 6(b). Figs. 6(c) and 6(d) show a closer view of the signatures where the improvement is more apparent.

## 6 Conclusions and Future Work

This paper brings an analysis of the error in the localization of edges detected by computing the zeros of the second derivative of an image in the direction of its gradient. A

closed-form expression for the derivative and second derivative of circularly symmetric pulses was presented along with experimental validation. These expressions were used to quantify the bias in the localization of the points of an edge as a function of the edge curvature, and these results were confirmed by experiments with synthetic data. Given the scale and edge curvature, this bias can then be corrected using the lookup table computed as described in this paper. This method of correction was validated using both synthetic and real images.

The techniques developed in this paper can be employed in the analysis of other problems in edge detection. A natural candidate would be the analysis of the bias in the localization of edges computed from the zero-crossings of the Laplacian. The Laplacian of any circularly symmetric pulse $\hat{s}(r)$, as in (8), is given by

$$\nabla^2 \hat{s}(r) = \frac{\partial^2 \hat{s}}{\partial r^2} + \frac{1}{r}\frac{\partial \hat{s}}{\partial r}. \tag{17}$$

Substituting (9) and (10) in (17), one easily obtains

$$\nabla^2 \hat{s}(r) = -\frac{R}{\sigma^2} e^{\frac{-(r^2+R^2)}{2\sigma^2}} \left[ \left( \frac{r}{\sigma^2} + \frac{2}{r} \right) I_1 \left( \frac{Rr}{\sigma^2} \right) - \frac{R}{\sigma^2} I_0 \left( \frac{Rr}{\sigma^2} \right) \right], \tag{18}$$

an expression which could be used to derive an analysis similar to the one carried out in this paper for the case of the Laplacian detector. Observe that this would be a significant improvement over the methodology used in [4], which provides a closed form solution for the Laplacian response of a disk image only for the central pixel of the pulse, and in [3,8], which model curved edges as two line segments meeting at an angle.

Another natural extension of this work is to evaluate the effect of neighboring structures on the localization of edges. Since convolution and derivatives are linear operations, an image composed of any linear combination of concentric pulses $s_i(r)$, each with its own radius $R_i$, would have first and second derivatives given by a sum of terms of the form of (9) and (10), respectively, and the techniques developed in this work could again be applied. The effect of noise on the shift in localization should also be studied.

The work presented here is primarily theoretical, and further experimentation on real images is needed. The application of this research to the correction of the underestimated area of tubular structures such as airways found by edge detection in CT images is ongoing.

## References

1. H. H. Baker and T. O. Binford. Depth from edge and intensity based stereo. In *Proc. of the 7th Int. Joint Conf. on Artificial Intell.*, pages 631–636, Vancouver, Canada, August 1981.
2. F. Bergholm. Edge focusing. *IEEE Trans. Pattern Analysis and Machine Intell.*, 9(6):726–741, November 1987.
3. V. Berzins. Accuracy of laplacian edge detectors. *Computer Vision, Graphics and Image Processing*, 27(2):195–210, August 1984.
4. D. Blostein and N. Ahuja. A multiscale region detector. *Computer Vision, Graphics and Image Processing*, 45(1):22–41, January 1989.
5. J. F. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, 8(6):679–698, November 1986.

6. J. J. Clark. Authenticating edges produced by zero-crossing algorithms. *IEEE Trans. Pattern Analysis and Machine Intell.*, 11(1):43–57, January 1989.
7. D. Cosandier and M. A. Chapman. High-precision target location for industrial metrology. In S. F. El-Hakim, editor, *Volumetrics II*, number 2067 in Proc. of SPIE, The Int. Soc. for Optical Engineering, pages 111–122, Boston, USA, September 1993.
8. R. Deriche and G. Giraudon. Accurate corner detection: An analytical study. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 66–70, Osaka, Japan, December 1990.
9. R. Deriche and G. Giraudon. A computational approach for corner and vertex detection. *Int. Journal of Computer Vision*, 10(2):101–124, April 1993.
10. J. H. Elder and S. W. Zucker. Local scale control for edge detection and blur estimation. *IEEE Trans. Pattern Analysis and Machine Intell.*, 20(7):699–716, July 1998.
11. O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. Artificial Intelligence Series. MIT Press, Cambridge, USA, 1993.
12. R. Kakarala and A. Hero. On achievable accuracy in edge localization. *IEEE Trans. Pattern Analysis and Machine Intell.*, 14(7):777–781, July 1992.
13. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *Int. Journal of Computer Vision*, 1(4):312–331, January 1988.
14. T. Lindeberg. *Scale-Space Theory in Computer Vision*, volume 256 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, USA, 1994.
15. E. Lyvers and O. Mitchell. Precision edge contrast and orientation estimation. *IEEE Trans. Pattern Analysis and Machine Intell.*, 10(6):927–937, November 1988.
16. D. Marr and E. Hildreth. Theory of edge detection. *Proc. Roy. Soc. London. B.*, 207:187–217, 1980.
17. D. Marr and T. Poggio. A computational theory of human stereo vision. *Proc. Roy. Soc. London B*, 204:301–328, 1979.
18. A. P. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Analysis and Machine Intell.*, 9(4):523–531, July 1987.
19. P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis and Machine Intell.*, 12(7):629–639, July 1990.
20. C. A. Rothwell, A. Zisserman, D. Forsyth, and J. L. Mundy. Fast recognition using algebraic invariants. In J. L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, Artificial Intelligence Series, chapter 20, pages 398–407. MIT Press, Cambridge, USA, 1992.
21. V. Torre and T. Poggio. On edge detection. *IEEE Trans. Pattern Analysis and Machine Intell.*, 8(2):147–162, March 1986.
22. L. J. van Vliet and P. W. Verbeek. Edge localization by MoG filters: Multiple-of-Gaussians. *Pattern Recognition Letters*, 15:485–496, May 1994.
23. P. W. Verbeek and L. J. van Vliet. On the location error of curved edges in low-pass filtered 2-D and 3-D images. *IEEE Trans. Pattern Analysis and Machine Intell.*, 16(7):726–733, July 1994.
24. G. N. Watson. *A Treatise on the Theory of Bessel Functions*. Cambridge University Press, Cambridge, UK, 1980 reprint of $2^{\mathrm{nd}}$ edition, 1922.
25. A. Zisserman, D. Forsyth, J. L. Mundy, and C. A. Rothwell. Recognizing general curved objects efficiently. In J. L. Mundy and A. Zisserman, editors, *Geometric Invariance in Computer Vision*, Artificial Intelligence Series, chapter 11, pages 228–251. MIT Press, Cambridge, USA, 1992.

# Texture Boundary Detection for Real-Time Tracking

Ali Shahrokni[1], Tom Drummond[2], and Pascal Fua[1⋆]

[1] Computer Vision Laboratory
EPFL
CH-1015 Lausanne, Switzerland
ali.shahrokni@epfl.ch
http://cvlab.epfl.ch
[2] Department of Engineering, University of Cambridge
Trumpington Street, Cambridge CB2 1PZ
http://www-svr.eng.cam.ac.uk/~twd20/

**Abstract.** We propose an approach to texture boundary detection that only requires a line-search in the direction normal to the edge. It is therefore very fast and can be incorporated into a real-time 3–D pose estimation algorithm that retains the speed of those that rely solely on gradient properties along object contours but does not fail in the presence of highly textured object and clutter.
This is achieved by correctly integrating probabilities over the space of statistical texture models. We will show that this rigorous and formal statistical treatment results in good performance under demanding circumstances

## 1 Introduction

Edge-based methods have proved very effective for fast 3–D model-driven pose estimation. Unfortunately, such methods often fail in the presence of highly textured objects and clutter, which produce too many irrelevant edges. In such situations, it would be advantageous to detect texture boundaries instead. However, because texture segmentation techniques require computing statistics over image patches, they tend to be computationally intensive and have therefore not been felt to be suitable for such purposes.

To dispel this notion, we propose a texture-based approach to finding the projected contours of 3–D objects while retaining the speed of standard edge-based techniques. We demonstrate its effectiveness for real-time tracking while using only a small fraction of the computational power a modern PC. Our technique is inspired by earlier work [1] on edge-based tracking that starts from the estimated projection of a 3–D object model and performs a line search in the direction perpendicular to the projected edges to find the most probable boundary location. Here, we replace conventional gradient-based edge detection by a method which can directly compute the most probable location for a texture boundary on the search line. To be versatile, the algorithm is designed to work even when neither of the textures on either side of the boundary are known *a priori*. Our technique is inspired by the use of Markov processes for texture description and segmentation. However, our requirements differ from those of classical texture segmentation methods

⋆ This work was supported in part by the Swiss National Science Foundation.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 566–577, 2004.
© Springer-Verlag Berlin Heidelberg 2004

in the sense that we wish to find the optimal pose parameters rather than arbitrary region boundaries.

This is challenging because speed requirements compel us to restrict ourselves to computing statistics along a line, and therefore a fairly limited number of pixels. We achieve this by correctly integrating probabilities over the space of statistical texture models. We will show that this rigorous and formal statistical treatment has allowed us to reach our goal under these demanding circumstances, which we regard as the main contribution of this paper. It is also worth noting that our implementation results in a real-time 3–D tracker that uses only a fraction of the computational resources of a modern PC, thus opening the possibility to simultaneously track many objects on ordinary hardware.

In the remainder of this paper, we first discuss related works. We then introduce our approach to detecting texture boundaries along a line. Finally, we integrate it into a contour-based 3–D tracker and demonstrate its effectiveness on real video-sequences.

## 2 Related Work and Background

We first briefly review the state-of-the art in real-time pose estimation and then discuss existing techniques for texture segmentation and show why they are not directly applicable to the kind of real-time processing we are contemplating in this paper.

### 2.1 Real-Time 3–D Tracking

Robust real-time tracking remains an open problem, even though offline camera registration from an image sequence [2,3,4] has progressed to the point where commercial solutions have become available. By matching natural features such as interest points between images these algorithms achieve high accuracy even without *a priori* knowledge. For example, in [3], the authors consider the image sequence hierarchically to derive robust correspondences and to distribute error over whole of it. Speed not being a critical issue, these algorithms take advantage of time-consuming but effective techniques such as bundle adjustment.

Model-based approaches, such as those proposed in [5,6], attempt to avoid drift by looking for a 3–D pose that correctly re-projects the features of a given 3–D model into the 2–D image. These features can be edges, line segments, or points. The best fit is found through least-squares minimisation of an error function, which may lead to spurious results when the procedure becomes trapped in erroneous local minima.

In earlier work [1], we developed such an approach that starts from the estimated projection of a 3–D object model and performs a line search in the direction perpendicular to the projected edges to find the most probable boundary location. Pose parameters are then taken to be those that minimise the distance of the model's projection to those estimated locations. This process is done in terms of the SE(3) group and its Lie algebra. This formulation is a natural choice since it exactly represents the space of poses that form the output of a rigid body tracking system. Thus it provides a canonical method for linearizing the relationship between image motion and pose parameters. The corresponding implementation works well and is very fast when the target object stands out clearly against the background. However it tends to fail for textured objects whose boundaries

are hard to detect unambiguously using conventional gradient-based techniques, as the chair shown in Fig. 1. We will argue that the technique proposed here remedies this failing using only a small fraction of the computational power of a modern PC. This is in contrast to other recent model-based techniques [7] that can also deal with textured objects but require the full power of the same PC to achieve real-time performance.

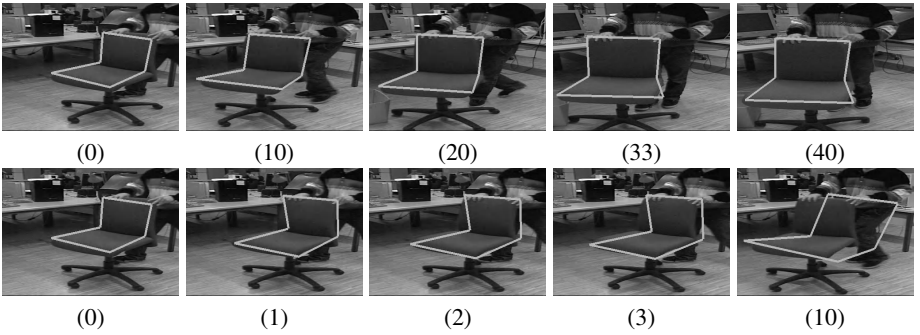



**Fig. 1.** Tracking a chair using a primitive model made of two perpendicular planes. Top row: Using the texture-based method proposed in this paper, the chair is properly tracked throughout the whole sequence. Bottom row: Using a gradient-based method to detect contours, the tracker starts being imprecise after the 3rd frame and fails completely thereafter.

### 2.2 Texture-Based Segmentation

Many types of texture descriptors have been developed to characterize textures. Gabor filters [8,9] have proved as an excellent descriptive tool for a wide range of textures [10] but are computationally exhaustive for real time applications. Multi-resolution algorithms offer some speed-up without loss of accuracy [11,12,10,13,14]. However, while this approach is effective for global segmentation of images containing complex textures, it is not the optimal solution for industrial applications in which the search space is limited. In the context of object tracking, the structure of the object to be tracked is already known, and there is a strong prior on its whereabouts. Hence, where real-time performance is required, classical texture segmentation is not the best choice, but a fast technique which localizes the target boundaries is desired.

However, the approach presented in this paper does borrow ideas from the texture segmentation literature. Hidden Markov random fields appear naturally in problems such as image segmentation, where an unknown class assignment has to be estimated from the observations at each pixel. Statistical modeling using Hidden Markov Models are very rich in mathematical structure and hence can form the theoretical basis for use in a wide range of applications. A comprehensive discussion about Markov Random Fields (MRF) and Hidden Markov Models (HMM) is given in [15,16]. [17] uses a Gibbs Markov Random Field to model texture which is fused with a 2D Gaussian distribution model for color segmentation for real time tracking. In contrast to HMM, MRF methods are non causal and therefore not desirable for line search methods for which a statistical model would be constructed during the search.

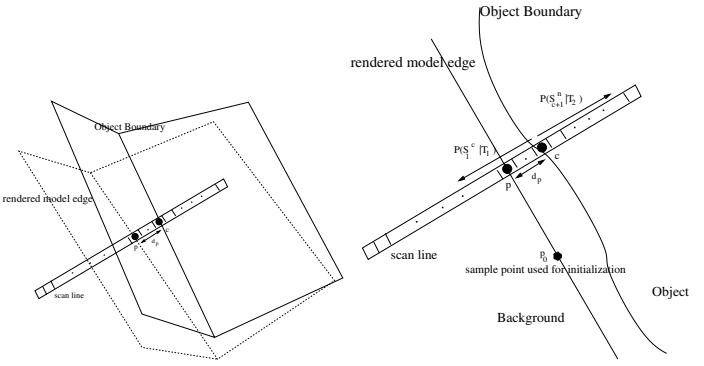



**Fig. 2.** Contour-based 3-D tracking. (a) Search for a real contour in the direction normal to projected edge.(b) Scanning a line through model sample point $p$ for which a statistical model associated to model point $p_0$ is retrieved and used to find the texture crossing point $c$. Notice that the statistical model of $p_0$ is made offline and conforms with the real object statistics in the neighbourhood.

## 3 Locating Texture Boundaries Using 1–D Line Search

As discussed in Section 2.1, we use as the basis for this work a tracker we developed in earlier work. Successive pose parameters are estimated by minimizing the observed distances from the rendered model to the corresponding texture crossing point, that is the point where the underlying statistics change, in the direction normal to the projected edge. This search is illustrated by Fig. 2. In this section, we formalize the criteria we use in this search and derive the algorithms we use to evaluate them.

A texture is modeled as a statistical process which generates a sequence of pixels. The problem is then cast as follows:A sequence of $n$ pixel intensities, $S_1^n = (s_1, ...s_n)$, is assumed to have been generated by two distinct texture processes each operating on either side of an unknown change point, as shown in Fig. 2(b). Thus the observed data is considered to have been produced by the following process: First a changepoint $c$ is selected uniformly at random from the range [1-n]. Then the pixels to the left of the changepoint (the sequence $S_1^c$) are produced by a texture process $T_1$ and the pixels to the right ($S_{c+1}^n$) are produced by process $T_2$. The task is then to recover $c$ from $S_1^n$. If both $T_1$ and $T_2$ are known then this corresponds to finding the $c$ that maximises:

$$P(\text{changepoint at } c|S_1^n, T_1, T_2) = KP(S_1^c|T_1)P(S_{c+1}^n|T_2) \ . \tag{1}$$

where $K$ is a normalisation constant. If one of the textures, for example, $T_1$ is unknown, then the term $P(S_1^c|T_1)$ must be replaced by the integral over all possible texture processes:

$$P(S_1^c) = \int P(S_1^c|T)P(T) \ \mathrm{d}T \ . \tag{2}$$

While it may be tempting to approximate this by considering only the most probable $T$ to have generated $S_1^c$, this yields a poor approximation for small data sets, such as

are exhibited in this problem. A key contribution of this paper is that we show how the integral can be solved in closed form for reasonable choices of the prior $P(T)$ (e.g. uniform).

In this work we consider two kinds of texture processes: first, one in which the pixel intensities are independently drawn from a probability distribution and second, one in which they are generated by a 1st order Markov process, which means that the probability of selecting a given pixel intensity depends (only) on the intensity of the preceding pixel. We refer to these two processes as 0 and 1st order models.

### 3.1 Solving for the 0th Order Model

The 0th order model states that the pixel intensities are drawn independently from a probability distribution over $I$ intensities ($T = \{p_i\}; i = 1..I$). If such a texture is known *a priori* then $P(S_1^c|T) = \prod_i p_{s_i}$. If the texture is unknown then:

$$P(S_1^c) = \int P(S_1^c|T)P(T)\ \mathrm{d}T = \int P(s_c|T)P(S_1^{c-1}|T)P(T)\ \mathrm{d}T \tag{3}$$

$$= P(S_1^{c-1}) \int p_{s_c} P(T|S_1^{c-1})\ \mathrm{d}T \tag{4}$$

The integral in (4) is $E(p_{s_c}|S_1^{c-1})$: i.e. the expected value of the probability $p_{s_c}$ in the texture given the observed sequence $S_1^{c-1}$. If we assume a uniform prior for $T$ over the $I$-1 simplex of probability distributions, then this integral becomes:

$$E(p_{s_c}|S_1^{c-1}) = \frac{\int_0^1 \int_0^{1-p_1} \cdots \int_0^{1-\sum_{i=1}^{I-2} p_i} \ p_{s_c} \prod_{j=1}^{I} p_j^{o_j} \ \ \mathrm{d}p_{I-1} \cdots \mathrm{d}p_2 \mathrm{d}p_1}{\int_0^1 \int_0^{1-p_1} \cdots \int_0^{1-\sum_{i=1}^{I-2} p_i} \prod_{j=1}^{I} p_j^{o_j} \ \ \mathrm{d}p_{I-1} \cdots \mathrm{d}p_2 \mathrm{d}p_1} \tag{5}$$

where there are $o_j$ occurrences of symbol $j$ in the sequence $S_1^{c-1}$. Note that both of these integrals have the same form, since the additional $p_{s_c}$ in the numerator can be absorbed into the product by adding one to $o_{s_c}$. Substituting $p_I = 1 - \sum_{i=1}^{I-1} p_i$ and repeatedly integrating by parts yields:

$$E(p_{s_c}|S_1^{c-1}) = \frac{o_{s_c} + 1}{c + I - 1} \tag{6}$$

This result states that if an unknown probability distribution is selected uniformly at random and a set of samples are drawn from this distribution, then the expected value of the distribution is the distribution obtained by adding one to the number of instances of each value observed in the sample set.

For example, if a coin is selected with a probability of flipping heads randomly drawn from the uniform distribution over [0,1], and it is flipped 8 times, giving 3 heads and 5 tails, then the probability that the next flip will be heads is (3+1)/(8+2) = 0.4.

This result can be applied recursively to the whole sequence to give Algorithm 1.

**Algorithm 1** Rapid 0th order computation of $\int P(S_1^c|T)P(T)\ \mathrm{d}T$

```
sequence_probability (S[], c)
  dim Observations[NUM_CLASSES]
  // seed Observations[] with 1 sample per bin
  for i=1..NUM_CLASSES do
    Observations[i]=1
  end for
  Probability=1
  for i=1..c do
    Probability = Probability * Observations[S[i]] / ∑ Observations[]
    Observations[S[i]] = Observations[S[i]]+1
  end for
  return Probability
```

### 3.2 Solving for the 1st Order Model

This idea can be immediately extended to a 1st order Markov process in which the intensities are drawn from a distribution which depends on the intensity of the preceding pixel ($T = \{p_{i|j}\}; i, j = 1..I$, where $p_{i|j}$ is the probability of observing intensity $i$ given that the previous pixel had intensity $j$). These $p_{i|j}$ can be considered as a transition matrix (row $i$, column $j$). Again, the probability of a sequence given a known texture is easy to compute:

$$P(S_1^c|T) = P(s_1|T)\prod_{i=2}^{c} p_{s_i|s_{i-1}} \quad \text{where } P(s_1|T) \text{ is using the 0th order model.} \tag{7}$$

For a first order Markov process, the 0th order statistics of the samples must be an eigenvector of $p_{i|j}$ with eigenvalue 1. Unfortunately, this means that a uniform prior for $T$ over $p_{i|j}$ is inconsistent with the uniform prior used in the 0th order case. To re-establish the consistency, it is necessary to choose a 1st order prior such that the expected value of a column of the transition matrix $p_{i|j}$ is obtained by adding $1/I$ rather than 1 to the number of observations in that column of the co-occurrence matrix before normalising the column to sum to 1. This means that the transition matrix is

$$E(p_{i|j}|S_1^c) = \frac{C_{ij} + 1/I}{1 + \sum_i C_{ij}} = \frac{C_{ij} + 1/I}{1 + o_j} \quad , \tag{8}$$

where $C_{ij}$ is the number of times that intensity $i$ follows intensity $j$ in the sequence $S_1^c$. And hence the expected 0th order distribution (which is the vector $\frac{(o_j+1)}{(c+I)}$) has the desired properties since

$$\sum_j E(p_{i|j}|S_1^c)\frac{(o_j+1)}{(c+I)} = \frac{\sum_j C_{ij} + 1/I}{c+I} = \frac{o_i+1}{c+I} \quad . \tag{9}$$

This modification is equivalent to imposing a prior over $p_{i|j}$ that favours structure in the Markov process and is proportional to $\prod_{ij} p_{i|j}^{(1/I-1)}$). This gives Algorithm 2.

**Algorithm 2** Rapid 1st order computation of $\int P(S_1^c|T)P(T)\ \mathrm{d}T$

```
sequence_probability (S[], c)
  dim CoOccurrence[NUM_CLASSES][NUM_CLASSES]
  // seed CoOccurrence[][] with 1/NUM_CLASSES samples per bin
  for r=1..NUM_CLASSES do
    for c=1..NUM_CLASSES do
      CoOccurrence[r][c]=1/NUM_CLASSES
    end for
  end for
  Probability=1/NUM_CLASSES      // probability of the first symbol
  for i=2..c do
    Probability = Probability * CoOccurrence[S[i]][S[i-1]] / ∑ CoOccurrence[][S[i-1]]
    CoOccurrence[S[i]][S[i-1]] = CoOccurrence[S[i]][S[i-1]]+1
  end for
  return Probability
```

### 3.3 Examples Applied to a Scanline through Brodatz Textures

We illustrate these ideas by considering the problem of locating the boundary between two Brodatz textures.

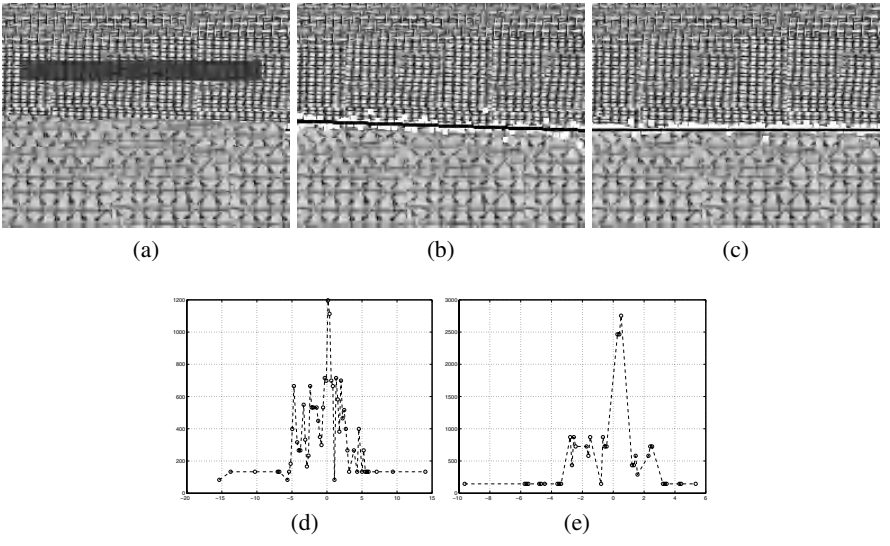



**Fig. 3.** Brodatz textures results. (a) texture patch used to learn the target texture model(the dark stripe). This model is used to detect the boundary of the target texture with another texture. (b) and (c) detected boundary using 0th and 1st order model respectively. White dots are the detected change point and the black line is the fitted texture boundary. (d)distances from edge in 0th order model (e)distances from edge in 1st order model

Fig. 3 shows detection results in a case where the texture is different at the top and bottom of the image. In Fig. 3 (a) the region that is used to generate model statistics is

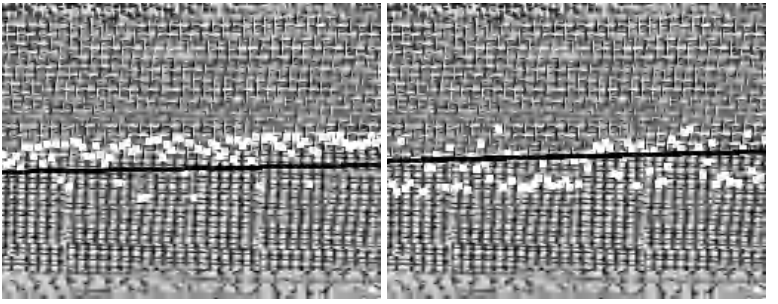
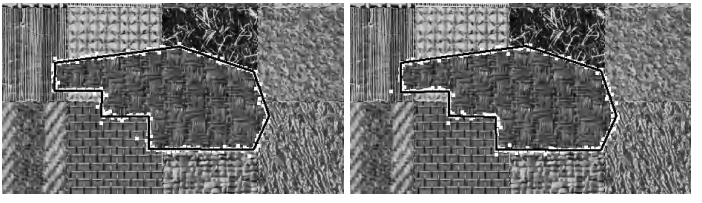


**Fig. 4.** In this case, the 0th order model (left) yields a result that is less precise than the 1st order one. As before, white dots are the detected change point and the black line is the fitted texture boundary.

marked by a dark stripe. In (b) the boundary between the upper and lower textures is correctly found using 0th order Markov model for both sides. (c) shows the boundary found using 1st order Markov model for both sides. While the model for the lower points is built in an autoregressive manner, the model of the upper points is the one created during the initialization phase (a). White dots show the exact detected point on the boundary along vertical scanlines, to which we robustly fit a line shown in black. The distribution of the distances from white points to the black line are shown in (d) and (e) for the 0th order and 1st order respectively. As can be noticed the distribution peak is less sharp for the 0th order model than the 1st order. This can hamper the detection of boundaries when the distributions are similar as shown in Fig. 4.

(a)



(b) (c)

**Fig. 5.** Segmentation of a polygonal patch. (a) Initialization: texture patch used to learn the target texture model(the dark stripe). This model is used to detect the boundary of the target texture with another texture. (b) and (c) detected boundary using 0 and 1st Markov model respectively. 0th order model is more sensitive to the initial conditions. As before, white dots are the detected change point and the black line is the fitted texture boundary.

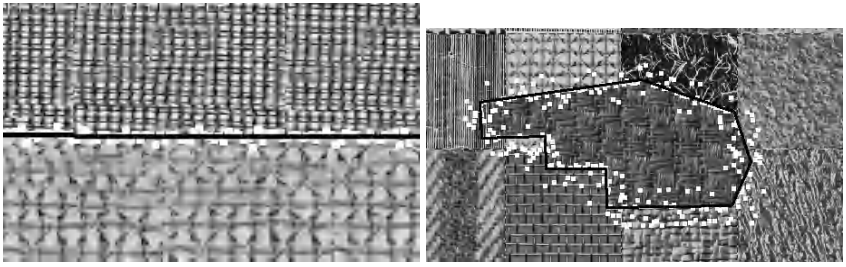


**Fig. 6.** Texture boundary detection with no a priori model assumption. In the case of polygonal texture, the white dots (detected changepoints are more scattered but robust fitting yields an accurate result for the boundary (black lines).

An example of texture segmentation for a piece of texture for which we have a geometric model is shown in Fig. 5. This is a especially difficult situation due to the neighbouring texture mixture. Nevertheless, both 0th and 1st order models detect the boundary of the target object (black lines) accurately by robust fitting of the polygonal model to the detected changepoints (white dots). However it was observed that 0th order model is more sensitive to initial conditions, which can be explained by the above statement that the 0th order observations are less accurate.

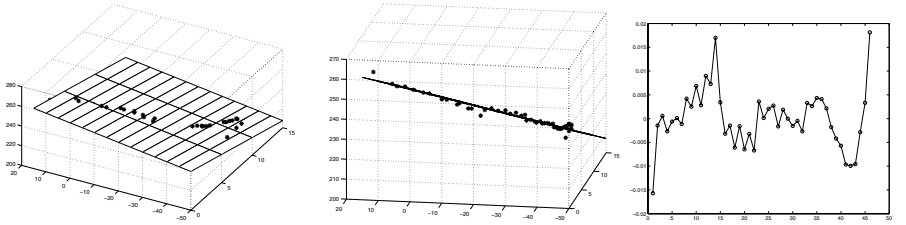
Our final test on textures involves the case where there is no a priori model for the texture on either side. Some results of application of our method are shown in Fig. 6. In the Fig. 6(a) the boundary is accurately detected. On the other hand for more complicated test of Fig. 6(b) the results are less accurate due to high outlier ratio.

## 4 Real-Time Texture Boundary Detection

We now turn to the actual real-time implementation of our texture-boundary detection method using the 0th an 1st order algorithms of Section 3.

Assuming a 1st order Markov model for the texture on both sides of the points along the search line, we wish to find the point for which the exterior and interior statistics best match their estimated model. These model can be built online or learned a priori. In the latter case, we build up and register a local 0th and 1st order probability distribution (intensity histograms and pixel intensity co-occurrence matrices) for a set of model sample points $M_0$ during the initialization where the $0$ subscript indicates time. Our system uses graphical rendering techniques to dynamically determine the visible edges and recognize the exterior ones. Our representation of the target model provides us with inward pointing normal vectors for its exterior edges. These normal vectors are used to read and store stripes of pixel values during initialization. The stripes need not be very large. These stripes are used to calculate local intensity histograms and pixel intensity co-occurrence matrices for each sample point in $M_0$. Having local information allows us to construct a more realistic model of texture in the cases where the target object contains patches of different texture.

During tracking, at time $i$, for each sample point $p$ in the set of sample $M_i$, we find the closest point in initialization sample set, $M_0$, and retrieve its associated histogram and co-occurrence matrix. Then, as depicted by Fig. 2(b), for each pixel $c$ along the



(a) (b) (c)

**Fig. 7.** Plotting the motion of the center of gravity of the chair of Fig. 1. (a,b) Top and side view of the plane fitted to the recovered positions of the center of gravity. (c) Deviations from the plane, which are very small (all measurements are in mm).

scanline associated to rendered point $p$, two probabilities $P(S^n_{c+1}|T_2)$ and $P(S^c_1|T_1)$ are calculated using algorithm 2 of Section 3, as the line is being scanned inwards in the direction of the edge normal. $P(S^n_{c+1}|T_2)$ is the probability that the successive pixels are part of the learned texture for the target object and $P(S^c_1|T_1)$ is the probability that the preceding pixels on the scanline are described by the the autoregressive model that is built as we move along the scan line and compute the integral of Eq. (2). This process is depicted by Fig. 2(b). This autoregeressive model is initially uniform and is updated for each new visited pixel and includes both the zeroth order distribution and a co-occurrence probability distribution matrix. Having calculated $P(S^n_{c+1}|T_2)$ and $P(S^c_1|T_1)$ for all points $c$ on the scanline, the point $c$ for which the total probability as given by Eq. (1) is maximum, is said to be the texture boundary and the distance $d_p$ between changepoint $c$ and sample point $p$ is passed to the minimiser to compute the pose parameters which minimises the total sum of distances.

## 5 Experimental Results

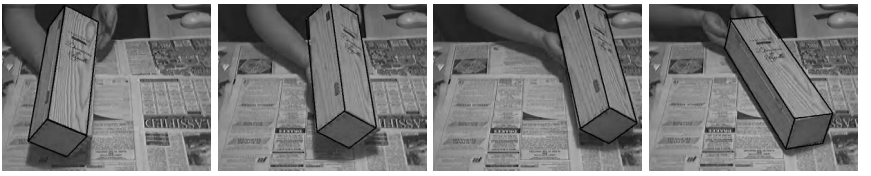
In the case of the chair of Fig. 1, everything else being equal, detecting the boundaries using the texture measure we propose appears to be much more effective that using gradients. To quantify this, in Fig. 7, we plot the motion of the center of gravity of the model recovered using the texture-based method. Since the chair remains on the ground, its true motion is of course planar but our tracker does not know this and has six degrees of freedom, three rotations and three translations. The fact that the recovered motion is also almost planar is a good indication that the tracking is quite accurate.

Figs. 8, 9, and 10 show the stable result of tracking different textured objects and an O2 computer against a cluttered background. Results of Fig. 9 are obtained without using prior models for the texture on either side of the model. Note that the algorithm works well on the O2 even though it is not particularly textured, showing its versatility.

Our current implementation can process up to 120 fps on a 2.6 GHz machine using a dense set of samples on our CAD model set and a 1st order statistical model. This low computational cost potentially allows the tracking of multiple textured and non-textured objects in parallel using a single PC.

(5) (36) (45) (101)

**Fig. 8.** Tracking a textured box against a cluttered background



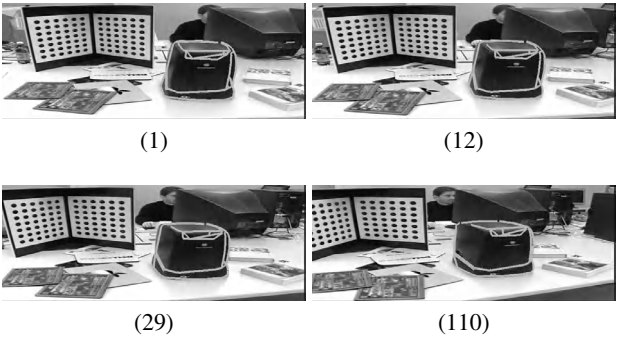
(35) (110) (190) (260)

**Fig. 9.** Tracking a textured box against a cluttered background without recording prior models. The model is materialised by black lines.

## 6 Conclusion

In this paper, we have shown that a well formalized algorithm based on a Markov model lets us use a simple line search to detect the transition from one texture to another that occurs at object boundaries. This results in a very fast technique that we have validated by incorporating it into a 3–D model-based tracker, which unlike those that rely on edge-gradients to detect contours, succeeds in the presence of texture and clutter.

We have demonstrated our technique's effectiveness to track man-made objects. However, all objects whose occluding contours can be estimated analytically are subject to this treatment. For example, many body tracking algorithms model the human body



(1) (12)

(29) (110)

**Fig. 10.** Tracking of a much less textured O2 computer against a cluttered background

as a set of cylinders or ellipsoids attached to an articulated skeleton. The silhouettes of these primitives have analytical expressions as a function of the pose parameters. It should therefore be possible to use the techniques proposed here to find the true outlines and deform the body models accordingly. This will be the focus of our future work in that area. Another issue for future work is the behaviour of this method when lighting conditions change. This will be handled by replacing the current stationary statistical models by dynamic ones that can evolve.

## References


1. Drummond, T., Cipolla, R.: Real-time tracking of highly articulated structures in the presence of noisy measurements. In: International Conference on Computer Vision, Vancouver, Canada (2001)
2. Tomasi, C., Kanade, T.: Shape and Motion from Image Streams under Orthography: A Factorization Method. International Journal of Computer Vision **9** (1992) 137–154
3. Fitzgibbon, A., Zisserman, A.: Automatic Camera Recovery for Closed or Open Image Sequences. In: European Conference on Computer Vision, Freiburg, Germany (1998) 311–326
4. Pollefeys, M., Koch, R., VanGool, L.: Self-Calibration and Metric Reconstruction In Spite of Varying and Unknown Internal Camera Parameters. In: International Conference on Computer Vision. (1998)
5. Marchand, E., Bouthemy, P., Chaumette, F., Moreau, V.: Robust real-time Visual Tracking Using a 2D-3D Model-Based Approach. In: International Conference on Computer Vision, Corfu, Greece (1999) 262–268
6. Lowe, D.G.: Robust model-based motion tracking through the integration of search and estimation. International Journal of Computer Vision **8(2)** (1992)
7. Vacchetti, L., Lepetit, V., Fua, P.: Fusing Online and Offline Information for Stable 3–D Tracking in Real-Time. In: Conference on Computer Vision and Pattern Recognition, Madison, WI (2003)
8. Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using gabor filters. Pattern Recognition **23(12)** (December 1991) 1167–1186
9. Bovik, A., Clark, M., Geisler, W.: Multichannel Texture Analysis Using Localized Spatial Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **12** (1990) 55–73
10. Puzicha, J., Buhmann, J.M.: Multiscale annealing for grouping and unsupervised texture segmentation. Computer Vision and Image Understanding: CVIU **76** (1999) 213–230
11. Bouman, C., Liu, B.: Multiple Resolution Segmentation of Textured Images. IEEE Transactions on Pattern Analysis and Machine Intelligence **13(2)** (1991) 99–113
12. Pietikinen, M., Rosenfeld, A.: Image Segmentation Using Pyramid Node Linking. IEEE Transactions on Systems, Man and Cybernetics **12** (1981) 822–825
13. Schroeter, P., Bigün, J.: Hierarchical Image Segmentation by Multi-dimensional Clustering and Orientation Adaptive Boundary Refinement. Pattern Recognition **28(5)** (1995) 695–709
14. Rubio, T.J., Bandera, A., Urdiales, C., Sandoval, F.: A hierarchical context-based textured image segmentation algorithm for aerial images. Texture2002 (http://www.cee.hw.ac.uk/ texture2002) (2002)
15. Li, S.Z.: Markov Random Field Modeling in Computer Vision. Springer-Verlag, Tokyo (1995)
16. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In: IEEE. Volume 77(2). (1989) 257 – 286
17. Ozyildiz, E.: Adaptive texture and color segmentation for tracking moving objects. Master's thesis, Pennsylvania State University (1999)

# A TV Flow Based Local Scale Measure for Texture Discrimination*

Thomas Brox and Joachim Weickert

Mathematical Image Analysis Group, Faculty of Mathematics and Computer Science,
Saarland University, Building 27, 66041 Saarbrücken, Germany
{brox,weickert}@mia.uni-saarland.de
www.mia.uni-saarland.de

**Abstract.** We introduce a technique for measuring local scale, based on a special property of the so-called total variational (TV) flow. For TV flow, pixels change their value with a speed that is inversely proportional to the size of the region they belong to. Exploiting this property directly leads to a region based measure for scale that is well-suited for texture discrimination. Together with the image intensity and texture features computed from the second moment matrix, which measures the orientation of a texture, a sparse feature space of dimension 5 is obtained that covers the most important descriptors of a texture: magnitude, orientation, and scale. A demonstration of the performance of these features is given in the scope of texture segmentation.

## 1 Introduction

Scale is an important aspect in computer vision, as features and objects are only observable on a certain range of scale. This is a consensus in the computer vision community for several years and has led to techniques that take into account multiple scales to reach their goal (scale spaces and multi-resolution approaches), or that try to automatically choose a good scale for their operators (scale selection).

As soon as *local* scale selection is considered, a measure for the local scale at each position in the image becomes necessary. The simplest idea is to consider the variance in a fixed local window to measure the scale. However, this has several drawbacks: large scales with high gradients result in the same value as small scales with low gradients. Moreover, the use of non-adaptive local windows always blurs the data. The latter drawback also appears with the general idea of local Lyaponov functionals [19], which includes the case of local variance. Other works on scale selection can be found in [10,11,12,5, 9]. All these methods have in common that they are gradient based, i.e. their measure of local scale depends directly on the local gradient or its derivatives. Consequently, the scale cannot be measured in regions without a significant gradient, and derivative filters of larger scale have to be used in order to determine the scale there. As non-adaptive, linear filters are used to represent these larger scales, this comes down to local windows and causes blurring effects. Although these effects are often hidden behind succeeding nonlinear operators like the maximum operator, some accuracy is lost here.

* Our research is partly funded by the project WE 2602/1-1 of the *Deutsche Forschungsgemeinschaft (DFG)*. This is gratefully acknowledged. We also want to thank Mikaël Rousson and Rachid Deriche for many interesting discussions on texture segmentation.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 578–590, 2004.
© Springer-Verlag Berlin Heidelberg 2004

In this paper we embark on another strategy that is not edge based but region based. Our local scale measure does not depend on the behavior of the gradient in scale space, but directly on the size of regions. This poses the question of how to define regions. In order to obtain an efficient technique, some special properties of total variation (TV) denoising, discovered recently in [4], are exploited. TV denoising [18] tends to yield piecewise constant, segmentation-like results, answering the question of how to define regions. Furthermore, it holds that pixels change their value inversely proportional to the size of the region they belong to. This directly results in a region based local scale measure for each pixel that does not need the definition of any window, and therefore yields the maximum localisation accuracy.

Note that there are applications for both edge based and region based methods. In the case of scale selection for edge detection, for instance, an edge based measure makes much more sense than a region based measure. For region based texture segmentation, however, it is exactly vice-versa.

The main motivation of why a region based local scale measure is important, can be found in the field of texture discrimination. There is a consistent opinion in the literature that Gabor filters [7] yield a good vocabulary for describing textures. Furthermore, findings in neurobiology indicate that some mechanism similar to Gabor filters is used in human vision [13]. A Gabor filter bank consists of a filter for each scale and each orientation. Unfortunately, this results in a large number of features that have to be integrated e.g. in a segmentation approach. As this may cause many problems, it has been proposed to reduce this over-complete basis by the general idea of sparse coding [14]. Another approach is to avoid the high dimensionality by using nearly orthogonal measures which extract the same features from the image, namely magnitude, orientation, and scale. An early approach to this strategy has been the use of the second moment matrix for texture discrimination in [2] and [15]. More recently, a texture segmentation technique based on the features of the second moment matrix coupled with nonlinear diffusion, the so-called nonlinear structure tensor, has been presented [17]. This publication demonstrated the performance of such a reduced set of features in texture discrimination. However, the second moment matrix only holds the information of the magnitude and orientation of a structure. The information of scale is missing. Consequently, the method fails as soon as two textures can only be distinguished by means of their scale. One can expect that a local scale measure would be very useful in this respect. Deriving such a measure for texture segmentation is the topic of the present paper.

**Paper organisation.** In the next section the new local scale measure based on TV flow is introduced. In Section 3 this measure will be coupled with the image intensity and the second moment matrix to form a new set of texture features. For demonstrating its performance in discrimination of textures, it will be used with the segmentation approach of [17]. We show experimental results and conclude the paper with a summary.

## 2 Local Scale Measure

In order to obtain a region based scale measure, an aggregation method is needed that determines regions. For this purpose, we focus on a nonlinear diffusion technique, the so-called *TV flow* [1], which is the parabolic counterpart to *TV regularisation* [18].

This diffusion method tends to yield piecewise constant, segmentation-like results, so it implicitly provides the regions needed for measuring the local scale. Starting with an initial image $I$, the denoised and simplified version $u$ of the image evolves under progress of artificial time $t$ according to the partial differential equation (PDE)

$$\partial_t u = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) \qquad u(t=0) = I. \tag{1}$$

The evolution of $u$ bit by bit leads to larger regions in the image, inside which all pixels have the same value. The goal is to measure the size of these regions.
Another useful property of TV flow, besides its tendency to yield segmentation-like results, is its linear contrast reduction [4]. This allows an efficient computation of the region sizes, without explicit computation of regions. Due to the linear contrast reduction, the size of a region can be estimated by means of the evolution speed of its pixels.
In 1D, space-discrete TV flow (and TV regularisation) have been proven to comply with the following rules [4]:

(i) A region of $m$ neighbouring pixels with the same value can be considered as one superpixel with mass $m$.

(ii) The evolution splits into merging events where pixels melt together to larger pixels.

(iii) Extremum pixels adapt their value to that of their neighbours with speed $\frac{2}{m}$.

(iv) The two boundary pixels adapt their value with half that speed.

(v) All other pixels do not change their value.

These rules lead to a very useful consequence: by simply sitting upon a pixel and measuring the speed with which it changes its value, it is possible to determine its local scale. As pixels belonging to small regions move faster than pixels belonging to large regions (iii), the rate of change of a pixel determines the size of the region it currently belongs to. Integrating this rate of change over the evolution time and normalising it with the evolution time $T$ yields the average speed of the pixel, i.e. its average inverse scale in scale space.

$$\frac{1}{m} = \frac{1}{2}\frac{\int_0^T |\partial_t u| \; dt}{T} \tag{2}$$

The integration has to be stopped after some time $T$, otherwise the interesting scale information will be spoiled by scale estimates stemming from heavily oversimplified versions of the image. The choice of $T$ will be discussed later in this section.
Since only extremum regions change their value, periods of image evolution in which a pixel is not part of such an extremum region have to be taken into account. This can be done by reducing the normalisation factor $T$ by the time at which the pixel does not move (v), leading to the following formula:

$$\frac{1}{m} = \frac{1}{2}\frac{\int_0^T |\partial_t u| \; dt}{\int_0^T (1 - \delta_{\partial_t u, 0}) \; dt} \tag{3}$$

where $\delta_{a,b} = 1$ if $a = b$, and 0 otherwise. Besides the estimation error for the boundary regions, where the scale is overestimated by a factor 2 (iv), this formula yields exact estimates of the region sizes without any explicit representation of the regions in the 1D case.

In 2D the topology of regions can become more complicated. A region can be an extremum in one direction and a saddle point in another direction. Therefore, it is no longer possible to obtain an *exact* estimate of the region sizes without an explicit representation of the regions. However, the extraction of regions is time consuming, and the formula in Eq. (3) still yields good approximations for the local scale in 2D, as can be seen in Fig. 1. Moreover, for texture discrimination, only a *relative* measure of the local scale is needed, and this measure will be combined with other texture features. When comparing two textured regions, the estimation error appears in both regions. For the case that the error is different in the two regions, they can be distinguished necessarily by the other texture features, since the topology of the two textures must be different. So the estimation error of the scale may not frustrate the correct distinction of two textured regions.

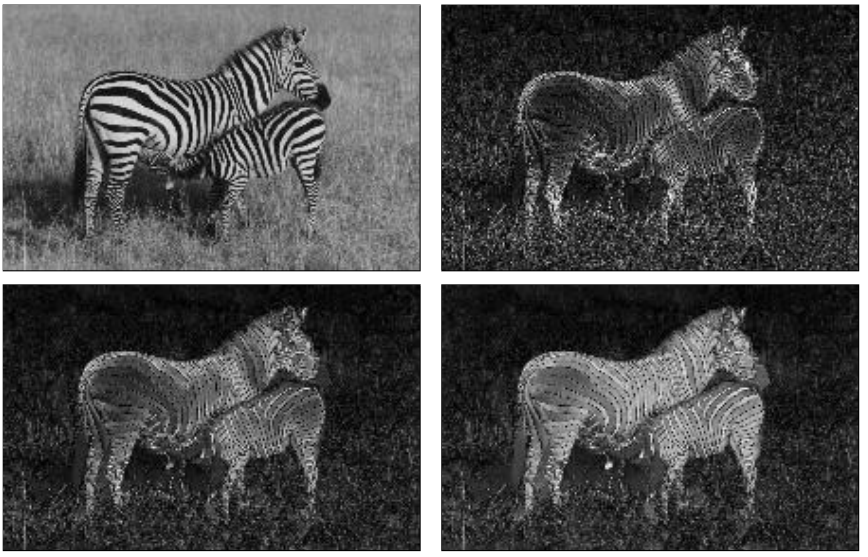


**Fig. 1.** TOP LEFT: (a) Zebra test image. TOP RIGHT: (b) Local scale measure with $T = 20$. BOTTOM LEFT: (c) $T = 50$. BOTTOM RIGHT: (d) $T = 100$. The scale measure yields the inverse scale, i.e. dark regions correspond to large scales, bright regions to fine scales.

On the first glance, it is surprising that there appears a scale parameter $T$ in the scale measure. On the other hand, it is easy to imagine that a pixel lives on several different scales during the evolution of an image $I$ from $t = 0$ to $t = T_{max}$, where the image

is simplified to its average value.[1] At the beginning, the pixel might be a single noise pixel that moves very fast until it is merged with other pixels to a small scale region. This small scale region will further move until it is merged with other regions to form a region of larger scale, and so on. Due to the integration, the average scale of the pixel's region during this evolution is measured, i.e. the complete history of the pixel up to a time $T$ is included. For texture discrimination one is mainly interested in the scale of the small scale texture elements, so it is reasonable to emphasise the smaller scales by stopping the diffusion process before $T_{max}$. [2]. Fig. 1 shows the local scale measure for different parameters $T$. In Fig. 1d the pixels of the grass texture have been part of the large background region for such a long time, that their small scale history has hardly any influence anymore. Here, mainly the stripes of the zebras stand out from the background. On the other hand, both results shown in Fig. 1b and Fig. 1c are good scale measures for small scale texture discrimination. Note that the diffusion speed of the pixels, i.e. the *inverse* scale, is computed, so dark regions correspond to large scales. For better visibility, the values have been normalised to a range between 0 and 255.

**Implementation.** TV flow causes stability problems as soon as the gradient tends to zero. Therefore the PDE has to be stabilised artificially by adding a small positive constant $\epsilon$ to the gradient (e.g. $\epsilon = 0.01$).

$$\partial_t u = \operatorname{div}\left(\frac{\nabla u}{\sqrt{u_x^2 + u_y^2 + \epsilon^2}}\right) \tag{4}$$

The stability condition for the time step size $\tau$ of an explicit Euler scheme is $\tau \leq 0.25\epsilon$, so for small $\epsilon$, many iterations are necessary. A much more efficient approach is to use a semi-implicit scheme such as AOS [20], which is unconditionally stable, so it is possible to choose $\tau = 1$. The discrete version of the scale measure for arbitrary $\tau$ is

$$\begin{aligned} u^0 &= I \\ u^{k+1} &= \frac{1}{2}\Big( \big((1) - 2\tau A_x(u^k)\big)^{-1} + \big((1) - 2\tau A_y(u^k)\big)^{-1} \Big) u^k \\ \frac{1}{m} &= \frac{1}{4\tau} \frac{\sum_{k=1}^{T} |u^{k+1} - u^k|}{\sum_{k=1}^{T} \left(1 - \delta_{(u^{k+1} - u^k),0}\right)} \end{aligned}$$

where $(1)$ denotes the unit matrix. $A_x$ and $A_y$ are the diffusion matrices in $x$ and $y$ direction (cf. [20]). The pre-factor $\frac{1}{4}$ instead of $\frac{1}{2}$ from Eq. (3) is due to the 2D case, where a pixel has 4 neighbours instead of 2.

[1] This finite extinction time $T_{max}$ is a special property of TV flow. An upper bound can be computed from the worst case scenario of an image with two regions of half the image size, maximum contrast $c_{max}$, and minimum boundary: $T_{max} \leq \frac{1}{4}\text{size}(I) \cdot c_{max}$.

[2] It would certainly be possible to automatically select an optimal scale $T$ where only large scale object regions remain in the image [12]. However, due to the integration, the parameter $T$ is very robust and for simplicity can be fixed at a reasonable value. For all experiments we used $T = 20$.

## 3 A Set of Texture Features

The local scale measure can be combined with the texture features proposed in [17]. Together, the four texture features and the image intensity can distinguish even very similar textures, as it will be shown in the next section.

Texture discrimination is a fairly difficult topic, since there is no clear definition, what texture is. Texture models can be distinguished into generative models and discriminative models. Generative models describe textures as a linear superposition of bases and allow to reconstruct the texture from its parameters. On the other side, discriminative models only try to find a set of features that allows to robustly distinguish different textures. A sound generative model has been introduced recently in [21]. A review of discriminative models can be found in [16]. While generative models are much more powerful in accurately describing textures, they are difficult to use for discrimination purposes so far.

The set of texture features we propose here, uses a discriminative texture model. It describes a texture by some of its most important properties: the intensity of the image, the magnitude of the texture, as well as its orientation and scale. These features benefit from the fact that they can be extracted also *locally*. Moreover, they can be used without further learning in any segmentation approach.

Orientation and magnitude of a texture are covered by the second moment matrix [6,15, 2]

$$J = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \tag{5}$$

yielding three different feature channels. Furthermore, the scale is captured by our local scale measure, while the image intensity is directly available. It should be noted that in contrast to many other features used for texture discrimination, all our features are completely rotationally invariant.

As proposed in [17], a coupled edge preserving smoothing process is applied to the feature vector. This coupled smoothing deals with outliers in the data, closes structures, and synchronises all channels, which eases further processing, e.g. in a segmentation framework. For a fair coupling it is necessary that all feature channels have approximately the same dynamic range. Furthermore, the normalisation procedure must not amplify the noise in the case that one channel shows only a low contrast. Therefore, only normalisation procedures that are independent of the contrast in the input data are applicable.

As a consequence, the second moment matrix is replaced by its square root. Given the eigenvalue decomposition $J = T(\lambda_i)T^\top$ of this positive semidefinite and symmetric matrix, the square root can be computed by

$$\tilde{J} := \sqrt{J} = T(\sqrt{\lambda_i})T^\top. \tag{6}$$

Since $J$ has eigenvalues $|\nabla I|^2$ and 0 with corresponding eigenvectors $\frac{\nabla I}{|\nabla I|}$ and $\frac{\nabla I^\perp}{|\nabla I|}$, this comes down to

$$\tilde{J} = \begin{pmatrix} \frac{I_x}{|\nabla I|} & -\frac{I_y}{|\nabla I|} \\ \frac{I_y}{|\nabla I|} & \frac{I_x}{|\nabla I|} \end{pmatrix} \begin{pmatrix} |\nabla I| & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{I_x}{|\nabla I|} & \frac{I_y}{|\nabla I|} \\ -\frac{I_y}{|\nabla I|} & \frac{I_x}{|\nabla I|} \end{pmatrix} = \begin{pmatrix} \frac{I_x^2}{|\nabla I|} & \frac{I_x I_y}{|\nabla I|} \\ \frac{I_x I_y}{|\nabla I|} & \frac{I_y^2}{|\nabla I|} \end{pmatrix} = \frac{J}{|\nabla I|}. \tag{7}$$

Using one-sided differences for the gradient approximation, the components of $\tilde{J}$ have the same dynamic range as the image $I$. With central differences, they have to be multiplied with a factor 2.
The range of the inverse scale $\frac{1}{m}$ is between 0 and 1, so after a multiplication with 255 (the maximum value of standard grey level images) all features have values that are bounded between 0 and 255.
In [17], TV flow was proposed for the coupled smoothing of the feature vector. Here, we use TV regularisation instead. In 1D, TV flow and TV regularisation yield exactly the same output [4]. In 2D, this equivalence could not be proven so far, however, both processes at least approximate each other very well. Hence, we consider the energy functional

$$E(u) = \int_\Omega \left( (u_1 - I)^2 + (u_2 - \tilde{J}_{11})^2 + (u_3 - \tilde{J}_{22})^2 + (u_4 - 2\tilde{J}_{12})^2 + (s \cdot u_5 - r)^2 + 2\alpha \sqrt{\sum_{k=1}^{5} |\nabla u_k|^2 + \epsilon^2} \right) dx$$

which consists of 5 data terms, one for each channel, and a coupled smoothness constraint that minimises the total variation of the output vector $u$. The values $r$ and $s$ are the numerator and denominator of the scale measure.

$$r := \frac{255}{4T} \int_0^T |\partial_t u| \, dt$$
$$s := \frac{1}{T} \int_0^T (1 - \delta_{\partial_t u, 0}) \, dt$$

The advantage of formulating the coupled smoothing as a regularisation approach is the so-called *filling-in effect* in cases where $r$ and $s$ are small, i.e. the confidence in the scale measure at this pixel is small. In such a case the term $(s \cdot u_5 - r)^2$ holding the scale measure gets small, so the smoothness term with the information of the other feature channels acquires more influence. Moreover, a division by 0 is avoided, if $r = s = 0$, what can happen when a pixel has never been part of an extremum region.
The Euler-Lagrange equations of this energy are given by

$$\begin{aligned} u_1 - I - \alpha \operatorname{div} (g \cdot \nabla u_1) &= 0 \\ u_2 - \tilde{J}_{11} - \alpha \operatorname{div} (g \cdot \nabla u_2) &= 0 \\ u_3 - \tilde{J}_{22} - \alpha \operatorname{div} (g \cdot \nabla u_3) &= 0 \\ u_4 - \tilde{J}_{12} - \alpha \operatorname{div} (g \cdot \nabla u_4) &= 0 \\ (s \cdot u_5 - r) \cdot s - \alpha \operatorname{div} (g \cdot \nabla u_5) &= 0 \end{aligned}$$

with $g := 1/\sqrt{\sum_{k=1}^{5} |\nabla u_k|^2 + \epsilon^2}$. They lead to a nonlinear system of equations, which can be solved by means of fixed point iterations and SOR in the inner loop. This has a similar efficiency as the AOS scheme used in [17] for TV flow. The texture features after the coupled smoothing are depicted in Fig. 2 and Fig. 3.

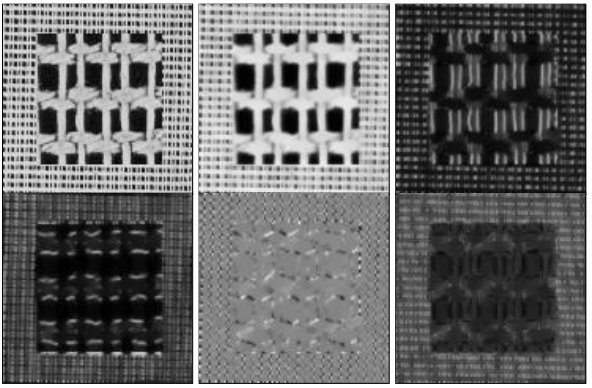


**Fig. 2.** FROM LEFT TO RIGHT, TOP TO BOTTOM: (a) Original image $I$ ($120 \times 122$). (b) Smoothed $I$ ($\alpha = 46.24$). (c) Smoothed $\tilde{J}_{11}$. (d) Smoothed $\tilde{J}_{22}$. (e) Smoothed $\tilde{J}_{12}$. (f) Smoothed scale measure (inverse scale).

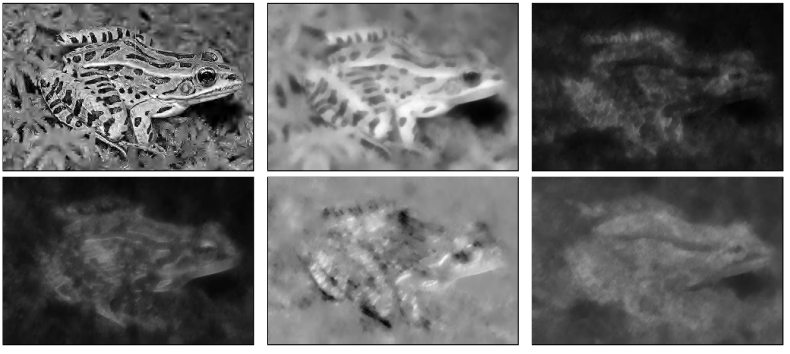


**Fig. 3.** FROM LEFT TO RIGHT, TOP TO BOTTOM: (a) Original image $I$ ($329 \times 220$). (b) Smoothed $I$ ($\alpha = 189.46$). (c) Smoothed $\tilde{J}_{11}$. (d) Smoothed $\tilde{J}_{22}$. (e) Smoothed $\tilde{J}_{12}$. (f) Smoothed scale measure (inverse scale).

## 4 Results

Before testing the performance of the texture features in a segmentation environment, we computed the dissimilarity between several textures from the Brodatz texture database [3]. As measure of dissimilarity for each texture channel we have chosen a simple distance measure taking into account the means $\mu_k(T)$ and the standard deviations $\sigma_k(T)$ of each feature channel $k$ of two textures $T_1$ and $T_2$:

$$\Delta_k = \left( \frac{\mu_k(T_1) - \mu_k(T_2)}{\sigma_k(T_1) + \sigma_k(T_2)} \right)^2 . \tag{8}$$

For the total dissimilarity the average of all 5 texture channels is computed:

$$\Delta = \frac{1}{5} \sum_{k=1}^{5} \Delta_k. \tag{9}$$

The resulting dissimilarities are shown in Fig. 4. The first value in each cell is the dissimilarity $\Delta_5$ according to the local scale measure only. The second value is the dissimilarity $\Delta$ taking all texture features into account. The computed values are in accordance with what one would expect from a measure of texture dissimilarity. Note that there are cases where $\Delta_5$ is significantly larger than $\Delta$. In such cases the local scale measure is very important to reliably distinguish the two textures.

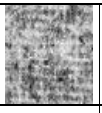
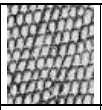
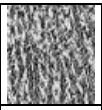
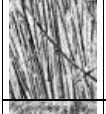
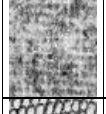
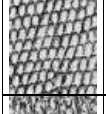
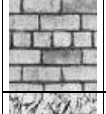
| | | | |  |  |  | | |
|---|---|---|---|---|---|---|---|---|
| | 0<br>**0** | 147<br>**114** | 399<br>**865** | 1388<br>**1514** | 1049<br>**379** | 23<br>**130** | 4449<br>**1893** | 735<br>**663** |
| | 147<br>**114** | 0<br>**0** | 92<br>**663** | 862<br>**1656** | 583<br>**470** | 60<br>**204** | 4174<br>**2258** | 331<br>**456** |
|  | 399<br>**865** | 92<br>**663** | 0<br>**0** | 340<br>**3068** | 199<br>**899** | 269<br>**565** | 2870<br>**3630** | 74<br>**846** |
|  | 1388<br>**1514** | 862<br>**1656** | 340<br>**3068** | 0<br>**0** | 11<br>**716** | 1235<br>**1443** | 1546<br>**421** | 78<br>**538** |
|  | 1049<br>**379** | 583<br>**470** | 199<br>**899** | 11<br>**716** | 0<br>**0** | 897<br>**274** | 1589<br>**923** | 27<br>**454** |
| | 23<br>**130** | 60<br>**204** | 269<br>**565** | 1235<br>**1443** | 897<br>**274** | 0<br>**0** | 4548<br>**2000** | 589<br>**663** |
|  | 4449<br>**1893** | 4174<br>**2258** | 2870<br>**3630** | 1546<br>**2107** | 1589<br>**923** | 4548<br>**2000** | 0<br>**0** | 1932<br>**983** |
| | 735<br>**663** | 331<br>**456** | 74<br>**846** | 78<br>**538** | 27<br>**454** | 589<br>**663** | 1932<br>**983** | 0<br>**0** |

**Fig. 4.** Dissimilarities measured between some Brodatz textures. First Value: Dissimilarity according to scale measure. Second Value: Dissimilarity according to all texture features.

It should be noted that in this simple experiment there have been no boundaries between textures, as the statistics have been computed separately for each texture. Therefore, this experiment shows only the global texture discrimination capabilities of the features,

when the texture regions are already known. In order to test the performance of the features when being applied to the much harder problem of discriminating textures with the region boundaries not known in advance, the features were incorporated into the segmentation technique described in [17]. Here, the good localisation accuracy of the features becomes crucial.

First the segmentation was applied to images which were chosen such that they mainly differ in scale. The results are depicted in Fig. 6 and Fig. 7. Note that the inner region of Fig. 7 is the downsampled version of the outer region so the regions solely differ in scale. Therefore, the segmentation fails, if the feature channel representing our scale measure is left out. It also fails, if this channel is replaced by three channels with responses of circular Gabor filters of different scale (see Fig. 5).

This shows, at least for texture segmentation, that the use of a direct scale measure outperforms the representation of scale by responses of a filter bank. Note that our scale measure is also more efficient, as the segmentation has to deal with less channels. The total computation time for the $220 \times 140$ zebra image on an Athlon XP 1800+ was 8 seconds, around 0.5 seconds to extract the local scale, 1.5 seconds to regularise the feature space, and 6 seconds for the segmentation.

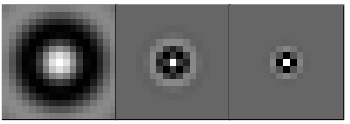


**Fig. 5.** Filter masks of the three circular Gabors used in the experiments.

Finally, the method was tested with two very challenging real world images. The results are depicted in Fig. 8 and Fig. 9. The fact that the method can handle the difficult frog image, demonstrates the performance of the extracted features used for the segmentation. Also the very difficult squirrel image, which was recently used in [8], could be segmented correctly, besides a small perturbation near the tail of the squirrel. Here again, it turned out that our scale representation compares favourably to circular Gabor filters.

## 5 Summary

In this paper, we presented a region based local scale measure. A diffusion based aggregation method (TV flow) is used to compute a scale space representing regions at different levels of aggregation. By exploiting the linear contrast reduction property of TV flow, the size of regions can be approximated efficiently without an explicit representation of regions. For each pixel and each level of aggregation the scale is determined by the size of the region the pixel belongs to. This scale is integrated over the diffusion time in order to yield the average scale for each pixel in scale space.

The local scale measure has been combined with other texture features obtained from the second moment matrix and the image intensity. Together, these features cover the most important discriminative properties of a texture, namely intensity, magnitude, orientation, and scale with only 5 feature channels. Consequently, a supervised learning stage to reduce the number of features is not necessary, and the features can be used directly with any segmentation technique that can handle vector-valued data. The performance in

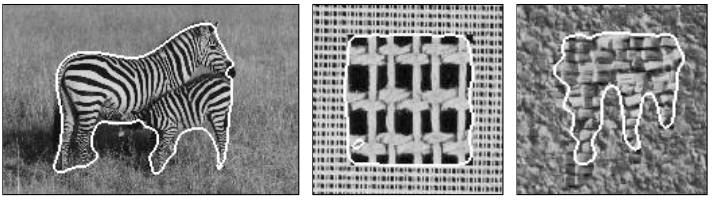


**Fig. 6.** Segmentation with the proposed texture features.

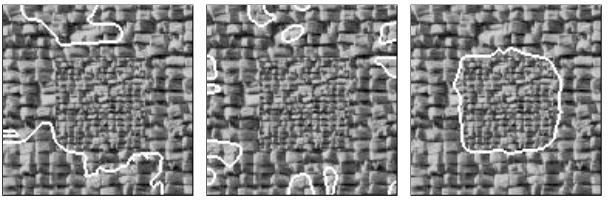


**Fig. 7.** LEFT: Segmentation without a scale measure. CENTER: Segmentation using 3 circular Gabors for scale. RIGHT: Segmentation using our local scale measure instead.

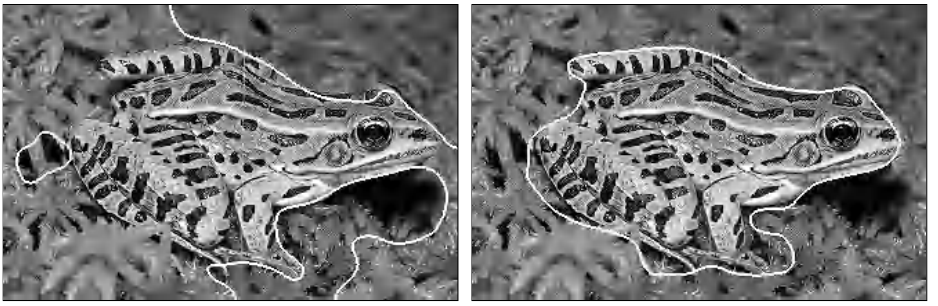


**Fig. 8.** LEFT: Segmentation using 3 circular Gabors for scale. RIGHT: Segmentation using our local scale measure instead.

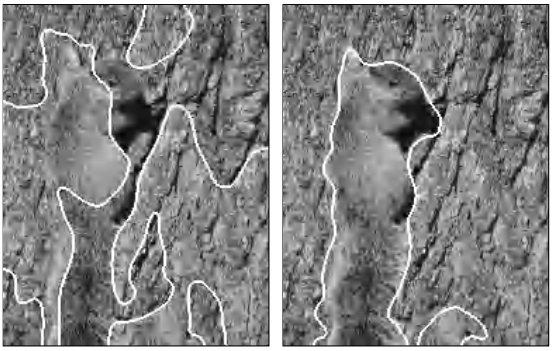


**Fig. 9.** LEFT: Segmentation using 3 circular Gabors for scale. RIGHT: Segmentation using our local scale measure instead.

texture discrimination has successfully been demonstrated by the segmentation of some very difficult texture images.

## References


1. F. Andreu, C. Ballester, V. Caselles, and J. M. Mazón. Minimizing total variation flow. *Differential and Integral Equations*, 14(3):321–360, Mar. 2001.
2. J. Bigün, G. H. Granlund, and J. Wiklund. Multidimensional orientation estimation with applications to texture analysis and optical flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):775–790, Aug. 1991.
3. P. Brodatz. *Textures: a Photographic Album for Artists and Designers*. Dover, New York, 1966.
4. T. Brox, M. Welk, G. Steidl, and J. Weickert. Equivalence results for TV diffusion and TV regularisation. In L. D. Griffin and M. Lillholm, editors, *Scale-Space Methods in Computer Vision*, volume 2695 of *Lecture Notes in Computer Science*, pages 86–100. Springer, Berlin, June 2003.
5. J. H. Elder and S. W. Zucker. Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716, July 1998.
6. W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.
7. D. Gabor. Theory of communication. *Journal IEEE*, 93:429–459, 1946.
8. M. Galun, E. Sharon, R. Basri, and A. Brandt. Texture segmentation by multiscale aggregation of filter responses and shape elements. In *Proc. IEEE International Conference on Computer Vision*, Nice, France, Oct. 2003. To appear.
9. G. Gómez, J. L. Marroquín, and L. E. Sucar. Probabilistic estimation of local scale. In *Proc. International Conference on Pattern Recognition*, volume 3, pages 798–801, Barcelona, Spain, Sept. 2000.
10. H. Jeong and I. Kim. Adaptive determination of filter scales for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(5):579–585, May 1992.
11. T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer, Boston, 1994.
12. T. Lindeberg. Principles for automatic scale selection. In B. Jähne, H. Haußecker, and P. Geißler, editors, *Handbook on Computer Vision and Applications*, volume 2, pages 239–274. Academic Press, Boston, USA, 1999.
13. S. Marcelja. Mathematical description of the response of simple cortical cells. *Journal of Optical Society of America*, 70:1297–1300, 1980.
14. B. A. Olshausen and D. J. Field. Sparse coding with an over-complete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.
15. A. R. Rao and B. G. Schunck. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53:157–185, 1991.
16. T. R. Reed and J. M. H. du Buf. A review of recent texture segmentation and feature extraction techniques. *Computer Vision, Graphics and Image Processing*, 57(3):359–372, May 1993.
17. M. Rousson, T. Brox, and R. Deriche. Active unsupervised texture segmentation on a diffusion based feature space. In *Proc. 2003 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 699–704, Madison, WI, June 2003. IEEE Computer Society Press.
18. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
19. J. Sporring, C. I. Colios, and P. E. Trahanias. Generalized scale-selection. Technical Report 264, Foundation for Research and Technology - Hellas, Crete, Greece, Dec. 1999.

20. J. Weickert, B. M. ter Haar Romeny, and M. A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, Mar. 1998.
21. S.-C. Zhu, C. Guo, Y. Wu, and W. Wang. What are textons? In A. Heyden, G. Sparr, M. NIelsen, and P. Johansen, editors, *Proc. 7th European Conference on Computer Vision*, volume 2353 of *Lecture Notes in Computer Science*, pages 793–807, Copenhagen, Denmark, May 2002. Springer.

# Spatially Homogeneous Dynamic Textures

Gianfranco Doretto, Eagle Jones, and Stefano Soatto

UCLA Computer Science Department
Los Angeles, CA 90095-1596
{doretto, eagle, soatto}@cs.ucla.edu

**Abstract.** We address the problem of modeling the spatial and temporal second-order statistics of video sequences that exhibit both spatial and temporal regularity, intended in a statistical sense. We model such sequences as dynamic multiscale autoregressive models, and introduce an efficient algorithm to learn the model parameters. We then show how the model can be used to synthesize novel sequences that extend the original ones in both space and time, and illustrate the power, and limitations, of the models we propose with a number of real image sequences.

## 1 Introduction

Modeling dynamic visual processes is a fundamental step in a variety of applications ranging from video compression/transmission to video segmentation, and ultimately recognition. In this paper we address a small but important component of the modeling process, one that is restricted to scenes (or portions of scenes) that exhibit both spatial and temporal regularity, intended in a statistical sense. Modeling visually complex dynamic scenes will require higher-level segmentation processes, which we do not discuss here, but having an efficient and general model of the spatio-temporal statistics at the low level is important in its own right.

While it has been observed that the distribution of intensity levels in natural images is far from Gaussian [1,2], the highly kurtotic nature of such a distribution is due to the presence of occlusions or boundaries delimiting statistically homogeneous regions. Therefore, *within such regions* it makes sense to employ the simplest possible model that can capture at least the second-order statistics. As far as capturing the temporal statistics, it has also been shown that linear Gaussian models of high enough order produce synthetic sequences that are perceptually indistinguishable from the originals, for sequences of natural phenomena that are well-approximated by stationary processes [3,4].

In this work, therefore, we seek to jointly model the spatio-temporal statistics of sequences of images that exhibit both spatial and temporal regularity using a simple class of dynamic multiscale autoregressive (MAR) models. We show how model parameters can be efficiently learned (iteratively, if the maximum likelihood solution is sought, or in closed-form if one can live with a sub-optimal estimate), and how they can be employed to synthesize sequences that extend in both space and time the original ones. This work builds on a number of existing contributions, which we summarize below.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 591–602, 2004.
© Springer-Verlag Berlin Heidelberg 2004

### 1.1 Relation to Previous Work

While extensive research has been focused on 2D texture analysis, we bypass this literature and focus only on that work which models textures in time. Such dynamic textures were first explored by Nelson and Polana [5], who extract spatial and temporal features from regions of a scene characterized by complex, non-rigid motion.

Szummer and Picard [6] use a spatio-temporal autoregressive model that creates local models of individual pixels based on causal neighbors in space and time. Bar-Joseph [7] employs multiresolution analysis of the spatial structure of 2D textures, and extends the idea to dynamic textures. Multiresolution trees are constructed for dynamic textures using a 3D wavelet transform; mutliple trees from the input are statistically merged to generate new outputs. This technique is unable to generate an infinite length sequence, however, as the trees span the temporal axis.

Video textures were developed by Schödl et al. [8], in which a transition model for frames of the original sequence is developed, alowing the original frames to be looped in a manner that is minimally noticeable to the viewer. Wei and Levoy [9] developed a technique in which pixels are generated by searching for a matching neighborhood in the original sequence.

In [3,4] we propose a simplified system identification algorithm for efficient learning of the temporal statistics, but no explicit model of the spatial statistics. Temporal statistics is also exploited by Fitzgibbon [10] to estimate camera motion in dynamic scenes. Recent work in texture synthesis includes a 2D and 3D patch-based approach by Kwatra et al. [11], using graph cuts to piece together new images and sequences. Wang et al. employ Gabor and Fourier decompositions of images, defining "movetons" [12], combined with statistical analysis of motion.

Our work aims at developing models of *both* the spatial and temporal statistics of a sequence that exploit statistical regularity in both domains. We seek the simplest class of models that achieve the task of capturing arbitrary second-order statistics, in the spirit of [4]. We achieve the goal within the MAR framework, and provide an efficient, closed-form learning algorithm to estimate the parameters of the model. We show this approach to be effective in capturing a wide variety of phenomena, allowing efficient description, compression, and synthesis.

## 2 Image Representation

In this section we introduce the class of multiscale stochastic models to be used in this paper, and describe how an image, that we model as a random field, can be represented in the multiscale framework. It has been shown that this framework can capture a very rich class of phenomena, ranging from one-dimensional Markov processes to $1/f$-like processes [13] and Markov random fields (MRFs) [14].

### 2.1 Multiscale Autoregressive Processes

The processes of interest are defined on a tree $\mathcal{T}$; we denote the nodes of $\mathcal{T}$ with an abstract index $s$. We define an upward shift operator $\gamma$ such that $s\gamma$ is the parent node of $s$. We consider regular trees where each node has $q$ children, and define a downward shift operator $\alpha$, such that the children of $s$ are indexed by $s\alpha_1, \ldots, s\alpha_q$ (see Fig. 1(a)). The nodes of the tree are organized in scales enumerated from 0 to $M$. The root node, $s = 0$, is the coarsest scale, while the finest scale consists of $q^M$ nodes. We indicate the scale of node $s$ with $d(s)$.

A multiscale autoregressive process $x(s) \in \mathbb{R}^{n(s)}$, $s \in \mathcal{T}$, is described via the scale-recursive dynamic model:

$$x(s) = A(s)x(s\gamma) + B(s)v(s) \quad , \tag{1}$$

under the assumptions that $x(0) \sim \mathcal{N}(0, P_0)$, and $v(s) \sim \mathcal{N}(0, I)$, where $v(s) \in \mathbb{R}^{k(s)}$ and $A(s)$ and $B(s)$ are matrices of appropriate size. The state variable $x(0)$ provides an initial condition for the recursion, while the driving noise $v(s)$ is white and independent of the initial condition.

Notice that model (1) is Markov from scale-to-scale. More importantly, any node $s$ on the $q$-adic tree can be viewed as a boundary between $q + 1$ subsets of nodes (corresponding to paths leading towards the parent and the $q$ offspring nodes). If we denote with $\Upsilon_1(s), \ldots, \Upsilon_q(s), \Upsilon_{q+1}(s)$, the corresponding $q+1$ subsets of states, the following property holds:

$$p(\Upsilon_1(s), \ldots, \Upsilon_q(s), \Upsilon_{q+1}(s)|x(s)) = p(\Upsilon_1(s)|x(s)) \cdots p(\Upsilon_{q+1}(s)|x(s)) \quad . \tag{2}$$

This property implies that there are extremely efficient and highly parallelizable algorithms for statistical inference [15], which can be applied to noisy measurements $y(s) \in \mathbb{R}^{m(s)}$ of the process given by:

$$y(s) = C(s)x(s) + w(s) \quad , \tag{3}$$

where $w(s) \sim \mathcal{N}(0, R(s))$ represents the measurement noise, and $C(s)$ is a matrix of appropriate size, specifying the nature of the process observations as a function of spatial location and scale.

### 2.2 Multiscale Representation of Images

We now describe how a Markov random field may be exactly represented in a multiscale framework. A thorough development of the material in this section may be found in [14].

Let us consider a regular discrete lattice $\Omega \in \mathbb{Z}^2$. The essence of the definition of a MRF $y(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$ is that there exists a neighborhood set $\Gamma_{\boldsymbol{x}}$, such that [16]

$$p(y(\boldsymbol{x})|\{y(\boldsymbol{z})|\boldsymbol{z} \neq \boldsymbol{x}\}) = p(y(\boldsymbol{x})|\{y(\boldsymbol{z})|\boldsymbol{z} \in \Gamma_{\boldsymbol{x}}\}) \quad . \tag{4}$$

For example, the *first-order* neighborhood of a lattice point consists of its four nearest neighbors, and the *second-order* neighborhood consists of its eight nearest neighbors.

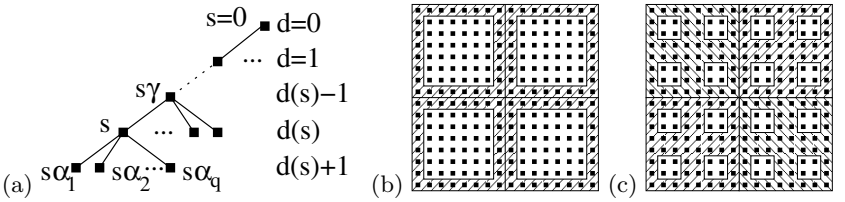



**Fig. 1.** (a) The state vector in a multiscale stochastic model is indexed by the nodes of a $q$-adic tree. The tree is a set of connected nodes rooted at 0. For a given node $s$, $s\gamma$ indexes the parent node of $s$, while $s\alpha_1, \ldots, s\alpha_q$ index its children. Finally, $d(s)$ indicates the scale of the node $s$. (b) The shaded region depicts the set $\Gamma(0)$ corresponding to the root of the tree for a $16 \times 16$ lattice. (c) To build the next level of the quad-tree for the multiscale representation, one proceeds recursively, defining $\Gamma(0\alpha_i)$, $i \in \{NW, NE, SE, SW\}$, where the subscripts refer to the spatial location (northwest, northeast, southeast, or southwest) of the particular child node.

We wish to use the multiscale framework to represent processes $y(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$ that are MRFs under second-order neighbors. If $\Omega$ is a lattice of $2^{M+2} \times 2^{M+2}$ points, a state at node $s$ on the $d$-th level of the tree is representative of the values of the MRF at $16(2^{M-d+1} - 1)$ points. We denote this set of points as $\Gamma(s)$. The shaded region in Fig. 1(b) depicts the set $\Gamma(0)$ corresponding to the root of the tree of a $16 \times 16$ lattice. Moreover, each set $\Gamma(s)$ can be thought of as the union of four mutually exclusive subsets of $4(2^{M-d+1} - 1)$ points, and we denote these subsets as $\Gamma_i(s)$, $i \in \{NW, NE, SE, SW\}$, where the subscripts refer to the spatial location of the subset. In Fig. 1(b), $\Gamma(0)$ is the union of the boundaries of four squares of $8 \times 8$ points, located in the northwest, northeast, southeast, and southwest quadrants of $\Omega$. To build the next level of the quad-tree for the multiscale representation, we proceed recursively, defining $\Gamma(0\alpha_i)$, $i \in \{NW, NE, SE, SW\}$. The points corresponding to the four nodes at depth $d = 1$ are shown in Fig. 1(c).

If we define $y(s) = \{y(\boldsymbol{x}) | \boldsymbol{x} \in \Gamma(s)\}$, and $y_i(s) = \{y(\boldsymbol{x}) | \boldsymbol{x} \in \Gamma_i(s)\}$, $i \in \{NW, NE, SE, SW\}$, from (4) it follows that

$$\begin{aligned} &p(y(s\alpha_{NW}), y(s\alpha_{NE}), y(s\alpha_{SE}), y(s\alpha_{SW}) | y(s)) = \\ &\quad p(y(s\alpha_{NW}) | y(s)) \cdots p(y(s\alpha_{SW}) | y(s)) = \\ &\quad p(y(s\alpha_{NW}) | y_{NW}(s)) \cdots p(y(s\alpha_{SW}) | y_{SW}(s)) \quad , \end{aligned} \tag{5}$$

and, given (2), it is straightforward that the MRF can be modelled as a multiscale stochastic process, and, in the Gaussian case, this leads to MAR models, given by equations (1) and (3).

## 3 A Representation for Space- and Time-Stationary Dynamic Textures

We seek a model that can capture the spatial and temporal "homogeneity" of video sequences that can also be used for extrapolation or prediction in both

space and time. We remind the reader at this stage that our work, unlike [8, 4], *cannot* capture scenes with complex layouts, due to the assumption of stationarity in space and time. In the experimental sections we show examples of sequences that obey the assumptions, as well as one that violates them.

We first observe that, depending on the modeling goals, a sequence of images $\{y(\boldsymbol{x},t)|t = 1,\dots,\tau,\boldsymbol{x} \in \Omega\}$, can be viewed as any of the following: (a) a collection of $\tau$ realizations of a stochastic process defined on $\Omega$; (b) a realization of a stochastic process defined in time; or (c) a realization of a stochastic process defined in both space and time. Let us consider the statistical description (for simplicity up to second-order), of $y(\boldsymbol{x},t)$, and define the mean of the process $m_y(\boldsymbol{x},t) \doteq E[y(\boldsymbol{x},t)]$, and the correlation function $r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2) \doteq E[y(\boldsymbol{x}_2,t_2)y^T(\boldsymbol{x}_1,t_1)]$. If in model (a) above we assume $m_y(\boldsymbol{x},t) = m_y \ \forall\ \boldsymbol{x},t$, $r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2) = r_y(\boldsymbol{x}_2-\boldsymbol{x}_1) \ \forall\ \boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2$, then the images are realizations from a stochastic process that is stationary in space; statistical models for such processes correspond to models for what are commonly known as planar 2D textures. If instead we take (b) and make the assumptions $m_y(\boldsymbol{x},t) = m_y(\boldsymbol{x}) \ \forall\ t$, $r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2) = r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_2-t_1) \ \forall\ t_1,t_2$, then the sequence of images is a realization of a stochastic process that is stationary in time. There are statistical models that aim to capture the time stationarity of video sequences, such as [4,12]. If in (c) we make the following assumptions:

$$m_y(\boldsymbol{x},t) = m_y \ \forall\, t,\boldsymbol{x}, \quad r_y(\boldsymbol{x}_1,\boldsymbol{x}_2,t_1,t_2) = r_y(\boldsymbol{x}_2-\boldsymbol{x}_1,t_2-t_1) \ \forall\, t_1,t_2,\boldsymbol{x}_1,\boldsymbol{x}_2, \tag{6}$$

then the sequence of images is a realization from a stochastic process that is stationary in time and space. Equation (6) represents the working conditions of this paper.

In order to model the second-order spatial statistics we consider the images $y(\boldsymbol{x},t)$, $t = 1,\dots,\tau$, as realizations from a single stationary Gaussian MRF, which can be represented as the output of a MAR model. That is,

$$\begin{cases} x(s,t) = A(s)x(s\gamma,t) + B(s)v(s,t) \\ y(s,t) = C(s)x(s,t) + w(s,t) \ , \end{cases} \tag{7}$$

where $x(0,t) \sim \mathcal{N}(0,P_0)$, $v(s,t) \sim \mathcal{N}(0,I)$, $w(s,t) \sim \mathcal{N}(0,R(s))$. Notice that, since the process is stationary in space, we have $A(s_1) = A(s_2)$, $B(s_1) = B(s_2)$, $C(s_1) = C(s_2), R(s_1) = R(s_2) \ \forall\ s_1,s_2 \in \mathcal{T}$, such that $d(s_1) = d(s_2)$. Therefore, with a slight abuse of notation, we now index every parameter previously indexed by $s$, with $d(s)$. For instance, $A(d(s)) \doteq A(s)$.

Employing model (7), an image at time $t$ is synthesized by independently drawing samples from the following random sources: $x(0,t)$ at the root of $\mathcal{T}$, and $v(s,t)$, for $s \in \mathcal{T} - \{0\}$. Since subsequent images of a video sequence exhibit strong correlation in time that is reflected as a temporal correlation of the random sources, we model $x(0,t)$ and $v(s,t)$ as the output of an autoregressive model. This means that, if we write $x(0,t) = B(0)v(0,t)$, $v(0,t) \sim \mathcal{N}(0,I)$, then the sources $v(s,t)$, $s \in \mathcal{T}$, evolve according to $v(s,t+1) = F(s,t)v(s,t) + u(s,t)$,

where $u(s,t) \sim \mathcal{N}(0, Q(s,t))$ is a white process. Also, since we model sequences that are stationary in time, we have $F(s,t) = F(s)$, and $Q(s,t) = Q(s), \forall t$.

Finally, given a sequence of images $\{y(\boldsymbol{x},t)|t = 1,\dots,\tau, \boldsymbol{x} \in \Omega\}$, satisfying conditions (6), we write the complete generative model as

$$\begin{cases} v(s,t+1) = F(s)v(s,t) + u(s,t) \\ x(0,t) = B(0)v(0,t) \\ x(s,t) = A(s)x(s\gamma,t) + B(s)v(s,t) \\ y(s,t) = C(s)x(s,t) + w(s,t) \ , \end{cases} \tag{8}$$

where $y(s,t) \in \mathbb{R}^{m(s)}$, $x(s,t) \in \mathbb{R}^{n(s)}$, $v(s) \in \mathbb{R}^{k(s)}$, $v(s,0) \sim \mathcal{N}(0,I)$; the driving noise $u(s,t) \sim \mathcal{N}(0,Q(s))$ and measurement noise $w(s,t) \sim \mathcal{N}(0,R(s))$ are independent across time and nodes of $\mathcal{T}$; and $A(s)$, $B(s)$, $C(s)$, $F(s)$, $Q(s)$, $R(s)$ are matrices of appropriate size.

## 4 Learning Model Parameters

Given a sequence of noisy images $\{y(\boldsymbol{x},t)|t = 1,\dots,\tau, \boldsymbol{x} \in \Omega\}$, we wish to learn the model parameters $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, $F(\cdot)$, along with the noise matrices $Q(\cdot)$ and $R(\cdot)$. This is a system identification problem [17] that can be posed as: given $y(\boldsymbol{x},1),\dots,y(\boldsymbol{x},\tau)$, find

$$\hat{A}(\cdot),\hat{B}(\cdot),\hat{C}(\cdot),\hat{F}(\cdot),\hat{Q}(\cdot),\hat{R}(\cdot) = \arg \max_{\substack{A(\cdot),B(\cdot),C(\cdot),\\ F(\cdot),Q(\cdot),R(\cdot)}} p(y(1),\dots,y(\tau)) \tag{9}$$

subject to (8). Problem (9) corresponds to the estimation of model parameters by maximizing the total likelihood of the measurements, and can be solved using an expectation-maximization (EM) procedure [18]. Note that the solution of problem (9) is not unique; in fact, once we choose the dimensions $n(s)$ and $k(s)$ of the hidden states $x(s,t)$ and $v(s,t)$, there is an equivalence class of models that maximizes the total likelihood. This class corresponds to the ensemble of every possible choice of basis for the states. Since we are interested in making broad use of model (8), while retaining its potential for recognition, we look for a learning procedure that selects one particular model from the equivalence class. To this end, we have designed a sub-optimal closed-form learning procedure, outlined in the following section.

### 4.1 A Closed-Form Solution

In this section we outline a procedure to choose a particular representative from the equivalence class of models that explains the measurements $\{y(\boldsymbol{x},t)|t = 1,\dots,\tau, \boldsymbol{x} \in \Omega\}$. Since describing the details of the procedure would require introduction of tedious notation, we retain only the most important ideas.

We first recall that $y(s,t) = \{y(\boldsymbol{x},t)|\boldsymbol{x} \in \Gamma(s)\}$, and define the set $\mathcal{Y}_{\mathcal{D}} \doteq \{y(s,t)|d(s) = \mathcal{D}\}$, where $\mathcal{D} = 0,\dots,M$, and $\Omega$ is a regular lattice of $2^{M+2} \times$

$2^{M+2}$ points. We will indicate with $Y_{\mathcal{D}}$ a matrix that contains in each column one element $y(s,t)$ of the set $\mathcal{Y}_{\mathcal{D}}$. For a given $t$ the measurements $y(s,t)$ are assumed to be lexicographically ordered in a column vector of $Y_{\mathcal{D}} \in \mathbb{R}^{m(s)\times 4^{\mathcal{D}}\tau}$, and we assume that the columns are suitably ordered in time. If we build the matrices $X_{\mathcal{D}}$ and $W_{\mathcal{D}}$ using $x(s,t)$ and $w(s,t)$, as we did with $Y_{\mathcal{D}}$, we note that $Y_{\mathcal{D}} = C(s)X_{\mathcal{D}} + W_{\mathcal{D}}\ \forall\ s$, such that $d(s) = \mathcal{D}$.

Since we work with images, we are interested in reducing the dimensionality of our data set, and we assume that $m(s) > n(s)$. Moreover, we seek a matrix $C(s)$ such that

$$C(s)^T C(s) = I\ . \tag{10}$$

These two assumptions allow us to fix the basis of the state space where $x(s,t)$ is defined, and eliminate the first source of ambiguity in the learning process. To see this, let us consider the estimation problem $\hat{C}(s), \hat{X}_{\mathcal{D}} = \arg\min_{C(s),X_{\mathcal{D}}} \|W_{\mathcal{D}}\|_F$, subject to (10). Take $Y_{\mathcal{D}} = U\Sigma V^T$ as the singular value decomposition (SVD) of $Y_{\mathcal{D}}$, where $U$ and $V$ are unitary matrices ($U^TU = I$ and $V^TV = I$). From the fixed rank approximation property of the SVD [19], the solution to our estimation problem is given by $\hat{C}(s) = U_{n(s)}$, and $\hat{X}_{\mathcal{D}} = \Sigma_{n(s)}V^T_{n(s)}$, where $U_{n(s)}$ and $V_{n(s)}$ indicate the first $n(s)$ columns of matrices $U$ and $V$, while $\Sigma_{n(s)}$ is a square matrix obtained by taking the first $n(s)$ elements of each of the first $n(s)$ columns of $\Sigma$. At this stage, if desired, it is also possible to estimate the covariance of the measurement noise. If $\hat{W}_{\mathcal{D}} = Y_{\mathcal{D}} - \hat{C}(s)\hat{X}_{\mathcal{D}}$, then the sample covariance is given by $\hat{R}(s) = \hat{W}^T_{\mathcal{D}}\hat{W}_{\mathcal{D}}/(4^{\mathcal{D}}\tau)$.

Now suppose that we are given $X_{\mathcal{D}} \in \mathbb{R}^{n(s)\times 4^{\mathcal{D}}\tau}$ and $X_{\mathcal{D}+1} \in \mathbb{R}^{n(s)\times 4^{\mathcal{D}+1}\tau}$; we are interested in learning $A(s)$, such that $d(s) = \mathcal{D}$. Note that the number of columns in $X_{\mathcal{D}+1}$ is four times the number of columns in $X_{\mathcal{D}}$, and the columns of $X_{\mathcal{D}+1}$ can be reorganized such that $X_{\mathcal{D}+1} = [X_{NW,\mathcal{D}+1}\ \ X_{NE,\mathcal{D}+1}\ \ X_{SE,\mathcal{D}+1}\ \ X_{SW,\mathcal{D}+1}]$, where $X_{i,\mathcal{D}+1}$, $i \in \{NW, NE, SE, SW\}$, represents the values of the state of the $i$-th quadrant. (For the remainder of this section we omit the specification of $i$ in the set $\{NW, NE, SE, SW\}$.) In this simplified version of the learning algorithm the rest of model parameters are actually sets of four matrices. In fact, $A(s)$ consists of a set of four matrices $\{A_i(S)\}$, that can easily be estimated in the sense of Frobenius by solving the linear estimation problem $\hat{A}_i(s) = \arg\min_{A_i(s)} \|X_{i,\mathcal{D}+1} - A_i(s)X_{\mathcal{D}}\|_F$. The closed-form solution can be computed immediately using the estimates $\hat{X}_{\mathcal{D}}$ and $\hat{X}_{i,\mathcal{D}+1}$ as: $\hat{A}_i(s) = \hat{X}_{i,\mathcal{D}+1}\hat{X}^T_{\mathcal{D}}(\hat{X}_{\mathcal{D}}\hat{X}^T_{\mathcal{D}})^{-1}$.

Once we know $A_i(s)$, we can estimate $B_i(s)$. In order to further reduce the dimensionality of the model, we assume that

$$n(s) > k(s)\ , \quad E[v_i(s,t)v_i(s,t)^T] = I\ . \tag{11}$$

These hypotheses allow us to fix the second source of ambiguity in the estimation of model parameters by choosing a particular basis for the state space where $v_i(s,t)$ is defined. Let us consider the SVD $X_{i,\mathcal{D}+1} - A_i(s)X_{i,\mathcal{D}} = U_v\Sigma_v V_v^T$, where $U_v$, and $V_v$ are unitary matrices. We seek $\hat{B}_i(s), \hat{V}_{i,\mathcal{D}} =$

$\arg\min_{B_i(s),V_{i,\mathcal{D}}} \|X_{i,\mathcal{D}+1} - A_i(s)X_{i,\mathcal{D}} - B_i(s)V_{i,\mathcal{D}}\|_F$, subject to (11), where $V_{i,\mathcal{D}}$ can be interpreted as $X_{i,\mathcal{D}}$. Again, from the fixed rank approximation property of the SVD [19], the solution follows immediately as: $\hat{B}_i(s) = U_{v,k(s)}S_{v,k(s)}/2^{\mathcal{D}}$, $\hat{V}_{i,\mathcal{D}} = 2^{\mathcal{D}}V^T_{v,k(s)}$, where $U_{v,k(s)}$ and $V_{v,k(s)}$ indicate the first $k(s)$ columns of the matrices $U_v$ and $V_v$, while $\Sigma_{v(s)}$ is a square matrix obtained by taking the first $k(s)$ elements of each of the first $k(s)$ columns of $\Sigma_v$.

Once we have $V_{i,\mathcal{D}}$, we generate two matrices, $V_{1,i,\mathcal{D}} \in \mathbb{R}^{k(s)\times 4^{\mathcal{D}}(\tau-1)}$ and $V_{2,i,\mathcal{D}} \in \mathbb{R}^{k(s)\times 4^{\mathcal{D}}(\tau-1)}$, with a suitable selection and ordering of the columns of $V_{i,\mathcal{D}}$ such that $V_{2,i,\mathcal{D}} = F_i(s)V_{1,i,\mathcal{D}} + U_{i,\mathcal{D}}$ and $U_{i,\mathcal{D}}$ has in its columns the values of $u_i(s,t)$. Then $F_i(s)$ can be determined uniquely, in the sense of Frobenius, by the solution to $\hat{F}_i(s) = \arg\min_{F_i(s)} \|U_{i,\mathcal{D}}\|_F$, given by $\hat{F}_i(s) = \hat{V}_{2,i,\mathcal{D}}\hat{V}^T_{1,i,\mathcal{D}}(\hat{V}_{1,i,\mathcal{D}}\hat{V}^T_{1,i,\mathcal{D}})^{-1}$. Finally, from the estimate $\hat{U}_{i,\mathcal{D}} = \hat{V}_{2,i,\mathcal{D}} - \hat{F}_i(s)\hat{V}_{1,i,\mathcal{D}}$, we learn the covariance of the driving noise $u_i(s,t)$: $\hat{Q}_i(s) = U_{i,\mathcal{D}}U^T_{i,\mathcal{D}}/4^{\mathcal{D}}(\tau-1)$.

## 5 A Model for Extrapolation in Space

While extrapolation or prediction in time is naturally embedded in the first equation of model (8), extrapolation in space outside the domain $\Omega$ is not implied because of the lack of the notion of *space causality*. Nevertheless, given that model (8) captures both the spatial and temporal stationarity properties of the measurements, the question of whether it is possible to extrapolate in space also becomes natural.

Given $y_i(0,t)$, $i \in \{NW, NE, SE, SW\}$, with the parameters of model (8), we can synthesize the random field inside $\Gamma_i(0)$. In the same way, if from the values taken by the MRF on $\Omega$, we predict the values $y_a(0,t)$ on the boundaries of the square $a$ (see Fig. 2(a)), then we are able to fill it in. Using this idea we could tile the whole 2D space. Since we already have a model for the spatial and temporal statistics of a single tile, we need only a model for the spatio-temporal prediction of the borders of the tiles. The model that we propose is very similar to model (8) when $s = 0$, and for the square $a$ it can be written as:

$$\begin{cases} v_a(0,t+1) = F_a(0)v_a(0,t) + u_a(0,t) \\ x_a(0,t) = A_a(0)x_\Omega(0,t) + B_a(0)v_a(0,t) \\ y_a(0,t) = C_a(0)x_a(0,t) + w_a(0,t) \ , \end{cases} \tag{12}$$

where the first and the last equations of the model, along with their parameters, have exactly the same meaning as the corresponding equations and parameters in model (8). The only difference is in the second equations of (8) and (12). While the meaning of $B_a(0)$ is the same as $B(0)$, (12) has an additional term $A_a(0)x_\Omega(0,t)$. With reference to Fig. 2(b), we take square $a$ as an example case. If we know the values taken by the MRF on the shaded half-plane, then from the properties of the MRF, only the values on the boundary (the vertical line) affect $y_a(0,t)$. In model (12) we make the further approximation that $y_a(0,t)$ is dependent only on the part of the boundary attached to it. Therefore, $x_\Omega(0,t)$ is

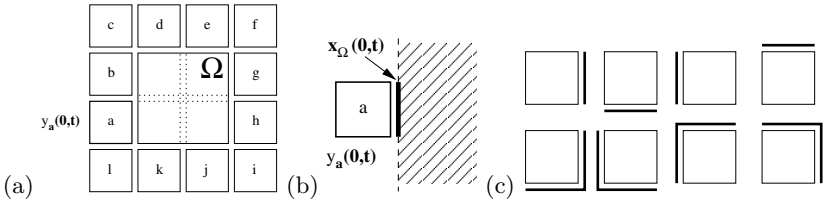



**Fig. 2.** Extrapolation in space. (a) Patches surrounding the original lattice $\Omega$ can be synthesized, if their borders can be inferred from the original data. (b) $y_a(0,t)$, the border of patch $a$, can be inferred from $x_\Omega(0,t)$, the portion of the original lattice which borders patch $a$. (c) The eight different ways in which borders must be inferred for a complete tiling of the 2D plane.

a state space representation of the bolded piece of the vertical line in Fig. 2(b). The matrix $A_a(0)$ can be learned from the training set in closed-form, in the sense of Frobenius, similarly to the manner in which we learned $A(s)$; the same is valid for the other parameters of the model. Fig. 2(c) represents the eight cases that need to be handled in order to construct a tiling of the 2D space in all directions.

## 6 Experiments

We implemented the closed-form learning procedure in Matlab® and tested it with a number of sequences. Figures 3(a)-(e) contain five samples from a dataset we created (boiling water, smoke, ocean waves, fountain, and waterfall). All the sequences have $\tau = 150$ RGB frames, and each frame is $128 \times 128$ pixels, so $M = 5$. These sequences exhibit good temporal and spatial stationarity properties. To improve robustness, we normalize the mean and variance of each sequence before running the algorithm. Fig. 3(f) depicts a sequence of fire from the MIT Temporal Texture database. This sequence has $\tau = 100$ RGB frames, with the same dimensions as the other sequences. These images are highly non-Gaussian, and, more importantly, exhibit poor stationarity properties in space. This sequence has been included to "push the envelope" and show what happens when conditions (6) are only marginally satisfied. For the sequences just described, the closed-form algorithm takes less than one minute to generate the model on a high-end (2 GHz) PC.

So far we have made the unspoken assumption that the order of the model $\{n(s), k(s), s \in \mathcal{T}\}$ was given. In practice, we need to solve a model selection problem based on the training set. The problem can be tackled by using information-theoretic or minimum description length criteria [20,21]. Following [22], we automate the selection by looking at the normalized energy encoded in the singular values of $\Sigma$ and $\Sigma_v$, and choose $n(s)$ and $k(s)$ so that we retain a given percentage of energy.

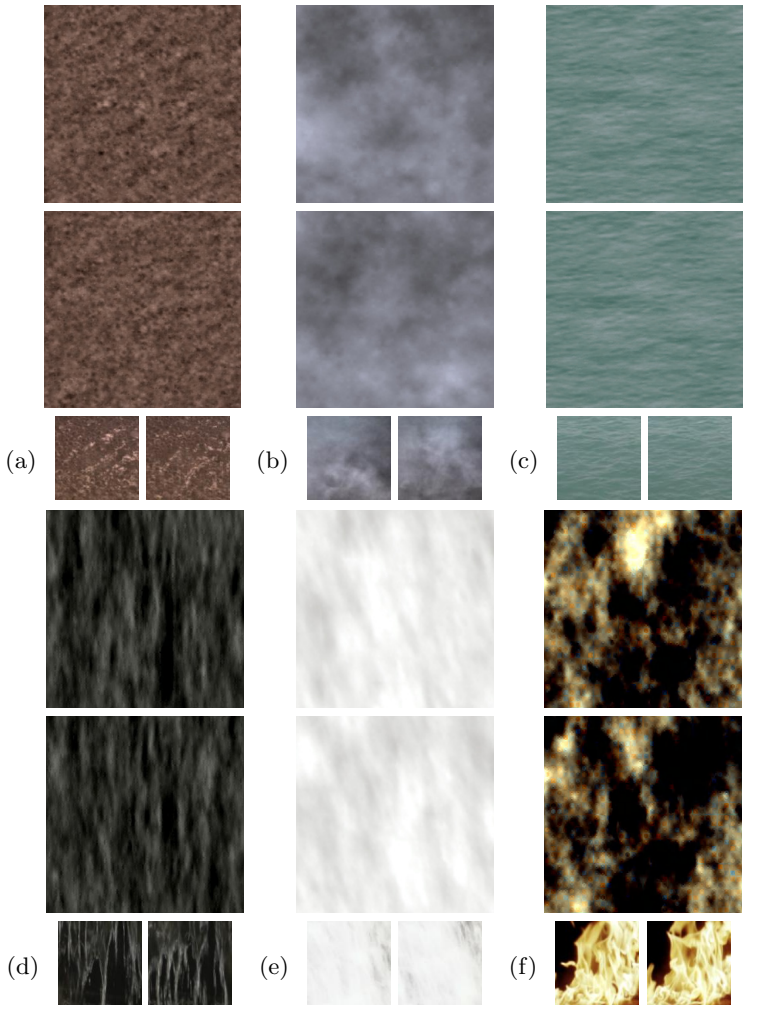



**Fig. 3.** (a)-(e) frames from five sequences which exhibit strong spatial and temporal regularity. Two frames of 128 × 128 pixels from the original sequences are shown on the bottom, and two synthesized frames (extended to 256 × 256 pixels using our spatial model) are shown on the top. Starting from top: (a) boiling water, (b) smoke, (c) ocean waves, (d) fountain, and (e) waterfall. (f) frames from a fire sequence with poor spatial regularity and non-Gaussian statistics. The frames synthesized by our technique correspond to a spatially "homogenized" version of the original sequence. All the data are available on-line at `http://www.cs.ucla.edu/~doretto/projects/dynamic-textures.html`.

### 6.1 Synthesis

Using model (8) to extrapolate in time, jointly with model (12) to extrapolate in space, we synthesized new sequences that are 300 frames long, with each frame

$256 \times 256$ pixels. Synthesis is performed by drawing random vectors $u(s,t)$ from Gaussian distributions, updating the states $v(s,t)$ and $x(s,t)$ and computing the pixel values. The process is implemented simply as matrix multiplications, and a $256 \times 256$ RGB image[1] takes as little as five seconds to synthesize. This computational complexity is relatively low, especially compared with MRF models that require the use of the Gibbs sampler. Besides being far less computationally expensive than such techniques, our algorithm does not suffer from the convergence issues associated with them.

Figures 3(a)-(e) show the synthesis results for five sequences. Our model captures the spatial stationarity (homogeneity) and the temporal stationarity of the training sequences[2]. We stress the fact that the training sequences are, of course, not perfectly stationary, and the model infers the "average" spatial structure of the original sequence.

Fig. 3(f) shows the synthesis results for a fire sequence. Since the spatial stationarity assumption is strongly violated, the model captures a "homogenized" spatial structure that generates rather different images from those of the training sequence[2]. Moreover, since the learning procedure factorizes the training set by first learning the spatial parameters ($C(s)$ and $A(s)$), and relies on the estimated state to infer the temporal parameters ($B(s)$ and $F(s)$), the temporal statistics (temporal correlation) appear corrupted if compared with the original sequence.

Finally, we bring the reader's attention to the ability of this model to compress the data set. To this end, we compare it with the classic dynamic texture model proposed in [4]. To a certain extent, this comparison is "unfair", because the classic dynamic texture model does not make any stationarity assumption in space. By making this assumption, however, we do expect to see substantial gains in compression. We generated the classic dynamic texture model, along with our model, for each training sequence, retaining the same percentage of energy in the model order selection step. We found that the ratio between the number of parameters in the classic model and our model ranged from 50 to 80.

## 7 Conclusion

We have presented a novel technique which integrates spatial and temporal modeling of dynamic textures, along with algorithms to perform learning from training data and generate synthetic sequences. Experimental results show that our method is quite effective at describing sequences which demonstrate temporal and spatial regularity. Our model provides for extension in time and space, and the implementation for both learning and synthesis is computationally inexpensive. Potential applications include compression, video synthesis, segmentation, and recognition.

[1] Multiple color channels are handled in the same manner as [4].

[2] Movies available on-line at `http://www.cs.ucla.edu/~doretto/projects/dynamic-textures.html`

**Acknowledgments.** This work is supported by AFOSR F49620-03-1-0095, NSF ECS-02–511, CCR-0121778, ONR N00014-03-1-0850:P0001.

## References

1. Huang, J., Mumford, D.: Statistics of natural images and models. In: Proc. CVPR. (1999) 541–547
2. Srivastava, A., Lee, A., Simoncelli, E., Zhu, S.: On advances in statistical modeling of natural images. J. of Math. Imaging and Vision **18** (2003) 17–33
3. Soatto, S., Doretto, G., Wu, Y.: Dynamic textures. In: Proc. ICCV, Vancouver (2001) 439–446
4. Doretto, G., Chiuso, A., Wu, Y., Soatto, S.: Dynamic textures. Int. J. Computer Vision **51** (2003) 91–109
5. Nelson, R., Polana, R.: Qualitative recognition of motion using temporal texture. CVGIP Image Und. **56** (1992) 78–89
6. Szummer, M., Picard, R.: Temporal texture modeling. In: Proc. ICIP. Volume 3. (1996) 823–826
7. Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., Werman, M.: Texture mixing and texture movie synthesis using statistical learning. IEEE Trans. Vis. Comp. Graphics **7** (2001) 120–135
8. Schödl, A., Szeliski, R., Salesin, D., Essa, I.: Video textures. In: Proc. SIGGRAPH. (2000) 489–498
9. Wei, L., Levoy, M.: Texture synthesis over arbitrary manifold surfaces. In: Proc. SIGGRAPH. (2001) 355–360
10. Fitzgibbon, A.: Stochastic rigidity: Image registration for nowhere-static scenes. In: Proc. ICCV, Vancouver (2001) 662–670
11. Kwatra, V., Schödl, A., Essa, I., Turk, G., Bobick, A.: Graphcut textures: Image and video synthesis using graph cuts. In: Proc. SIGGRAPH. (2003) 277–286
12. Wang, Y., Zhu, S.: A generative method for textured motion: Analysis and synthesis. In: Proc. ECCV. (2002) 583–598
13. Daniel, M., Willsky, A.: The modeling and estimation of statistically self-similar processes in a multiresolution framework. IEEE Trans. Inf. T. **45** (1999) 955–970
14. Luettgen, M., Karl, W., Willsky, A., Tenney, R.: Multiscale representations of markov random fields. IEEE Trans. Sig. Proc. **41** (1993)
15. Chou, K., Willsky, A., Beveniste, A.: Multiscale recursive estimation, data fusion, and regularization. IEEE Trans. Automat. Contr. (1994) 464–478
16. Geman, S., Geman, D.: Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Trans. PAMI **6** (1984) 721–741
17. Ljung, L.: System Identification - Theory for the User. Prentice Hall, Englewood Cliffs, NJ (1987)
18. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. J. R. Stat. Soc. B **39** (1977) 185–197
19. Golub, G., Van Loan, C.: Matrix Computations, Third Ed. Johns Hopkins Univ. Press, Baltimore, MD (1996)
20. Lütkepohl, H.: Introduction to Time Series Analysis. Second edn. Springer-Verlag, Berlin (1993)
21. Rissanen, J.: Modeling by shortest data description. Automatica **14** (1978) 465–471
22. Arun, K., Kung, S.: Balanced approximation of stochastic systems. SIAM J. on Matrix Analysis and Apps. **11** (1990) 42–68

# Synthesizing Dynamic Texture with Closed-Loop Linear Dynamic System

Lu Yuan[1,2⋆], Fang Wen[1], Ce Liu[3⋆], and Heung-Yeung Shum[1]

[1] Visual Computing Group, Microsoft Research Asia, Beijing 100080, China,
[2] Department of Electronic Engineering, Tsinghua University, Beijing 100084, China,
[3] CSAIL, Massachusetts Institute of Technology, Cambridge, 02139, USA
yuanl02@mails.tsinghuat.edu.cn, {t-fangw,hshum}@microsoft.com
celiu@mit.edu

**Abstract.** Dynamic texture can be defined as a temporally continuous and infinitely varying stream of images that exhibit certain temporal statistics. Linear dynamic system (LDS) represented by the state-space equation has been proposed to model dynamic texture[12]. LDS can be used to synthesize dynamic texture by sampling the system noise. However, the visual quality of the synthesized dynamic texture using noise-driven LDS is often unsatisfactory. In this paper, we regard the noise-driven LDS as an open-loop control system and analyze its stability through its pole placement. We show that the noise-driven LDS can produce good quality dynamic texture if the LDS is oscillatory. To deal with an LDS not oscillatory, we present a novel approach, called closed-loop LDS (CLDS) where feedback control is introduced into the system. Using the succeeding hidden states as an input reference signal, we design a feedback controller based on the difference between the current state and the reference state. An iterative algorithm is proposed to generate dynamic textures. Experimental results demonstrate that CLDS can produce dynamic texture sequences with promising visual quality.

## 1 Introduction

Dynamic texture, also known as temporal texture or video texture, can be defined as a temporally continuous and infinitely varying stream of images that exhibit certain temporal statistics. For example, it could be a continuously gushing fountain, a flickering fire, an escalator, and a smoke puffing slowly from the chimney – each of these phenomena possesses an inherent dynamics that a general video may not portray. Dynamic texture can be used for many applications such as personalized web pages, screen savers and computer games.

Dynamic texture analysis and synthesis have recently become an active research topic in computer vision and computer graphics. Similar to 2D texture synthesis, the goal of dynamic texture synthesis is to generate a new sequence that is similar to, but somewhat different from, the original video sequence.

⋆ This work was done when the authors worked in Microsoft Research Asia.

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3022, pp. 603–616, 2004.
© Springer-Verlag Berlin Heidelberg 2004

Ideally the generated video can be played endlessly without any visible discontinuities. Moreover, it is desirable that dynamic texture be easily edited and be efficient for storage and computation.

Most recent techniques on dynamic texture synthesis can be categorized into nonparametric and parametric methods. The nonparametric methods generate dynamic textures by directly sampling original pixels[15], frames[11], wavelet-structures[1], 2D patches[8] and 3D voxels[8] from a training sequence and usually produce high-quality visual effects. Compared with the nonparametric approaches, the parametric methods provide better model generalization and understanding of the essence of dynamic texture. The typical parametric models include Szummer and Picard's STAR model[13], Wang and Zhu's movetons representation[14], and Soatto et al's linear dynamic system (LDS) model[12]. They are very helpful for the tasks such as dynamic texture editing[4], recognition[10], segmentation[3] and image registration[5]. However, most parametric models are less likely to generate dynamic textures with the same quality as the nonparametric models, in particular for the videos of natural scenes.

In this paper, we argue that a carefully designed parametric model can compete well with any existing nonparametric model, e.g. [8], in terms of visual quality of the synthesized dynamic texture. Our work is inspired by the former work of noise-driven LDS for dynamic texture synthesis [12]. From a viewpoint of control theory, the noise-driven LDS is indeed an open-loop control system which is likely to be contaminated by noise. Based on the analysis of the stability of the open-loop LDS, we find that it is the problems of the pole placement and model fitting error that prevent the previous approaches from generating satisfactory dynamics textures.

In consequence, we propose a novel closed-loop LDS (CLDS) approach to synthesizing dynamic texture. In our approach, the difference between the reference input and the synthesized output is computed as a feedback control signal. Using a feedback controller, the whole closed-loop system can minimize the model fitting error and improve the pole placement. Our experimental results demonstrate that our approach can produce visually promising dynamic texture sequences.

## 2 Dynamic Texture Synthesis Using LDS

Under the hypothesis of temporal stationarity, Soatto et al[12] adopted a linear dynamic system (LDS) to analyze and synthesize dynamic texture. The state-space representation of their model is given by

$$\begin{cases} \boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + \boldsymbol{v}_t, \quad \boldsymbol{v}_t \sim \mathcal{N}(0, \Sigma_v) \\ \boldsymbol{y}_t = C\boldsymbol{x}_t + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, \Sigma_w) \end{cases} \tag{1}$$

where $\boldsymbol{y}_t \in \mathbb{R}^n$ is the observation vector; $\boldsymbol{x}_t \in \mathbb{R}^r, r \ll n$ is the hidden state vector; $A$ is the system matrix; $C$ is the output matrix and $\boldsymbol{v}_t, \boldsymbol{w}_t$ are Gaussian white noises.

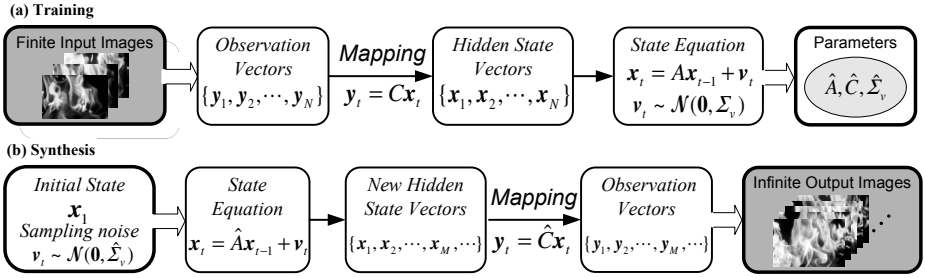



**Fig. 1. Framework of dynamic texture analysis and synthesis using LDS.** (a) In the training process, a finite sequence of input images is used to train an LDS model with the system matrix $A$, the observation matrix $C$ and the system noise $\boldsymbol{v}_t$ (with its variance $\Sigma_v$). (b) In the synthesis process, a new (possibly infinite) sequence is generated using the learnt LDS and by sampling system noise.

Fig. 1 illustrates the dynamic texture analysis and synthesis framework using the above LDS. To train the LDS model, a finite sequence of images $\{I_t\}_{t=1}^N$ subtracted by the mean image are concatenated to column vectors as observation vectors $\{\boldsymbol{y}_t\}_{t=1}^N$. These observation vectors are mapped into hidden state vectors $\{\boldsymbol{x}_t\}_{t=1}^N$ in a lower dimensional space by SVD analysis. We use these hidden states to fit a linear dynamic system and identify the parameters of the system through maximum likelihood estimation (MLE). The learnt LDS model is then used to synthesize a new sequence. Given an initial hidden state $\boldsymbol{x}_1$, the sampling noise $\boldsymbol{v}_t \sim \mathcal{N}(0, \hat{\Sigma}_v)$ can drive the system matrix $\hat{A}$ to generate new state vectors, which are then mapped to the high-dimensional observation vectors to form a new sequence of dynamic texture.

The above dynamic texture model has a firm analytic footing in system identification. However, it often fails to produce satisfactory results for simple sequences such as those shown in Fig. 7(b)(f). An open question is: how well one can synthesize a new sequence from the learnt LDS.

## 3 Analysis of Open-Loop LDS

According to control theory, the noise-driven dynamic texture synthesis process ($\boldsymbol{x}_t = A\boldsymbol{x}_{t-1} + \boldsymbol{v}_t$) illustrated in Fig. 1(b) can be regarded as a basic open-loop LDS, as shown in Fig. 3(a). The stability of an open-loop LDS is determined by its poles[6].

The poles of the system are exactly defined as the eigenvalues of system matrix $A$ [9], which take complex values. In other words, the location of the pole on the 2D pole plot (i.e. the polar coordinate system) is determined by the corresponding eigenvalue. If the magnitude of an eigenvalue is less than one, the pole is inside the unit circle in the pole plot, called a *stable pole* since this eigen-component will vanish eventually. If the magnitude equals one, the pole is on the unit circle, called an *oscillatory pole* because the magnitude of this

eigen-component will remain constant while the phase may change periodically. If the magnitude is greater than one, the pole is outside the unit circle, called an *unstable pole* because this eigen-component will magnify to infinity. Clearly, a control system cannot contain any unstable poles.

We consider the state equation $\boldsymbol{x}_t = A\boldsymbol{x}_{t-1}$ ignoring the noise term, where $A \in \mathbb{R}^{r\times r}$. By eigenvalue decomposition, we have $A = Q\Lambda Q^{-1}$, where $\Lambda = \text{diag}\{\sigma_1, \sigma_2, \ldots, \sigma_r\}$ and $Q \in \mathbb{R}^{r\times r}$. $\{\sigma_i\}_{i=1}^r$ are the eigenvalues of $A$, namely poles. Given an initial state vector $\boldsymbol{x}_1$, the state vector at time $t$ is given by

$$\boldsymbol{x}_t = A^{t-1}\boldsymbol{x}_1 = Q\Lambda^{t-1}Q^{-1}\boldsymbol{x}_1 = Q \cdot \text{diag}(\sigma_1^{t-1}, \cdots, \sigma_r^{t-1}) \cdot Q^{-1}\boldsymbol{x}_1.$$

For any eigenvalue or pole $\sigma_i$ we have

$$\lim_{t\to\infty} \left|\sigma_i^{t-1}\right| = \begin{cases} 0 & |\sigma_i| < 1, \\ 1 & |\sigma_i| = 1, \\ \infty & |\sigma_i| > 1. \end{cases}$$

There are three interesting cases with different pole distributions.

(a) All the poles of the dynamic system are stable, i.e. $\forall i \in \{1, 2, \cdots, r\} : |\sigma_i| < 1$. Then $\lim_{t\to\infty} \|\boldsymbol{x}_t\|_2 = 0$, where $\|\boldsymbol{x}_t\|_2$ represents the energy of the state vector. All the stable poles contribute to the decay of the dynamic system. This system is called a *stable system* (in accordance to the name of poles).

(b) Some poles are oscillatory and the rest are stable, i.e. $\forall i : |\sigma_i| \leqslant 1$ and $\exists j : |\sigma_j| = 1$. Then $\lim_{t\to\infty} \|\boldsymbol{x}_t\|_2 = c$, where $c$ is a constant. We call it an *oscillatory system*.

(c) There exist unstable poles, namely $\exists i : |\sigma_i| > 1$. Then $\lim_{t\to\infty} \|\boldsymbol{x}_t\|_2 = \infty$. The unstable poles contribute to the divergence of the dynamic system. With such an *unstable system*, one cannot generate an infinite sequence.

These cases are further illustrated by dynamic texture synthesis results from three typical sequences.

(a') The poles of the FOUNTAIN sequence are all stable (Fig. 2(d)). Thus, the generated dynamic sequence decays gradually as shown in Fig. 2(a).

(b') Since most poles of the ESCALATOR sequence are oscillatory and the rest are stable (Fig. 2(e)), the whole dynamic system is oscillatory and the generated sequence forms a satisfactory loop as shown in Fig. 2(b).

(c') The FIRE sequence has two unstable poles (Fig. 2(f)), which lead to the divergence of the dynamic system. This explains why the intensity of the synthesized fire saturates as shown in Fig. 2(c).

The key observation from the synthesis and synthesized examples above is that for dynamic texture synthesis, the learnt LDS must be an *oscillatory system*. Unfortunately, such a requirement on the learnt system matrix $A$ is too demanding to be satisfied in practice.

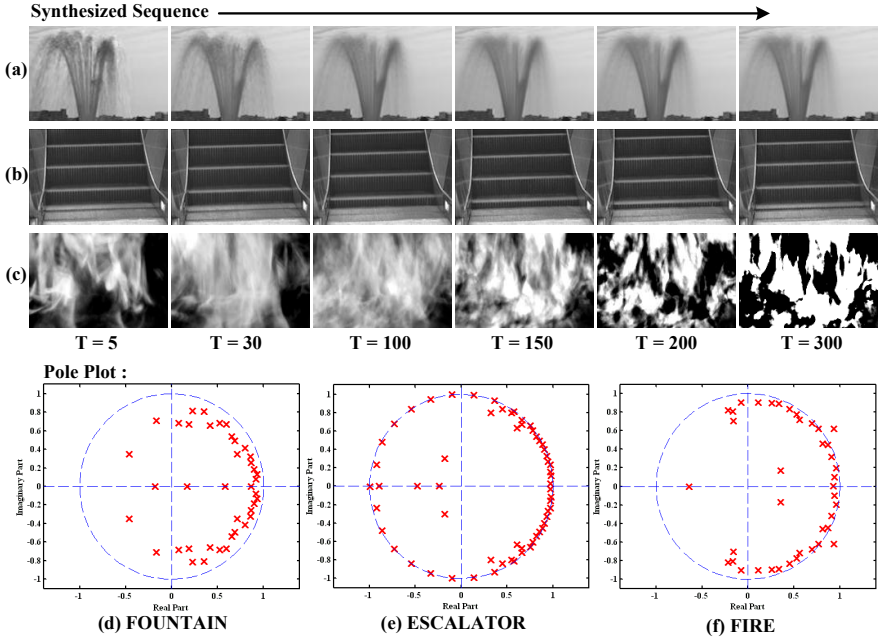



**Fig. 2. Effect of poles of system:** (a)(b)(c) show the synthesized sequences at different times. (a) Decaying FOUNTAIN sequence. (b) Oscillatory ESCALATOR sequence. This is the desirable result for dynamic texture. (c) Divergent FIRE sequence. (d) The positions of all the FOUNTAIN's poles (denoted by red crosses) are inside the unit circle. (e) Most poles of ESCALATOR are on the unit circle and others are inside the unit circle. (f) Two poles of FIRE are outside the unit circle and others are within or on the unit circle.

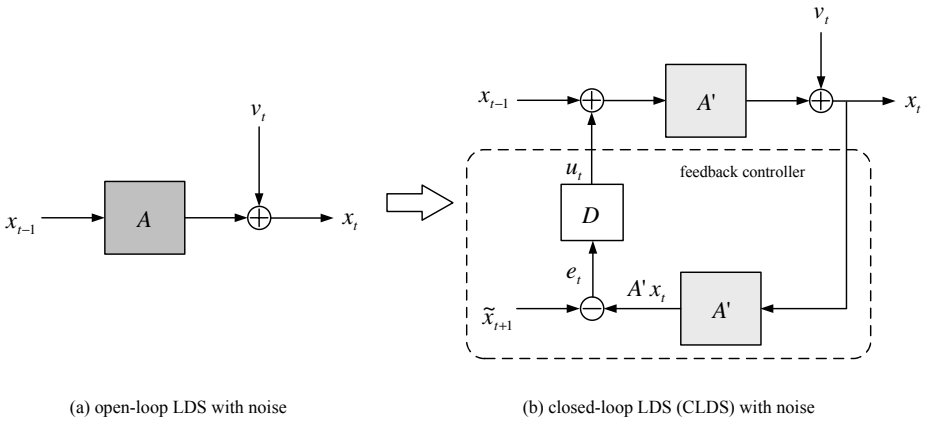
## 4 Closed-Loop LDS

If the learnt LDS from the input video sequence is not an oscillatory system, we propose to change the open-loop LDS to a closed-loop LDS (CLDS), as illustrated in Fig. 3(b), so that we can still synthesize dynamic texture with good visual quality. The overall goal of feedback control is to cause the output variable of a dynamic process to follow a desired reference variable accurately, regardless of any external disturbance in the dynamics of the process[7].

### 4.1 A Simple CLDS

The key component in the proposed CLDS is the definition of the error term which turns to be the control signal, given as

$$\begin{cases} \boldsymbol{x}_t = A'\boldsymbol{x}_{t-1} + A'\boldsymbol{u}_t + \boldsymbol{v}_t, \quad v_t \sim \mathcal{N}(0, \Sigma_v) \\ \boldsymbol{u}_t = D\boldsymbol{e}_t \\ \boldsymbol{e}_t = \tilde{\boldsymbol{x}}_{\boldsymbol{t+1}} - A'\boldsymbol{x}_t \\ \boldsymbol{y}_t = C\boldsymbol{x}_t + \boldsymbol{w}_t, \quad \boldsymbol{w}_t \sim \mathcal{N}(0, \Sigma_w) \end{cases} \tag{2}$$




(a) open-loop LDS with noise (b) closed-loop LDS (CLDS) with noise

**Fig. 3. Comparison of frameworks between open-loop LDS and closed-loop LDS:** In the figure, only the state-space representation is illustrated. In both models, $\boldsymbol{x}_t$ represents the hidden state at time $t$ in low-dimensional state-space in two frameworks; $\boldsymbol{v}_t$ represents the noise of the system. $A'$ represents the system matrix of our system while $A$ is the system matrix of open-loop LDS. $\boldsymbol{u}_t$ is the control signal; $D$ is the control matrix in our system and $\boldsymbol{e}_t$ is the difference between reference input and output.

where $\boldsymbol{y}_t$ is the observation vector; $\boldsymbol{x}_t$ is the state vector; $C$ is the output matrix; $\boldsymbol{e}_t$ is the difference between the reference and the rough estimate of the next state; $D$ is the proportional control matrix of the error term and $A'$ is the system matrix. In the state-space equation above, the current state $\boldsymbol{x}_t$ comprises three parts: 1) the rough estimation of the current state, i.e. $A'\boldsymbol{x}_{t-1}$, 2) the control signal $\boldsymbol{u}_t$ proportional to the difference between the reference input for next state $\tilde{\boldsymbol{x}}_{\boldsymbol{t+1}}$ and the rough estimation of the next state $A'\boldsymbol{x}_t$ from the output current state $\boldsymbol{x}_t$, and 3) sampled noise $\boldsymbol{v}_t$.

Clearly the open-loop LDS discussed previously (Fig. 3(a)) is a special case of the proposed CLDS (Fig. 3(b)) when $D$ is zero.

### 4.2 A Practical Model

The CLDS above is of the simplest form, a first-order system and a proportional controller. In practice, estimation of the current state vector $\boldsymbol{x}_t$ using the first-order model is often insufficient. Instead, we employ a number of previous state vectors $\{\boldsymbol{x}_{t-p}, \cdots, \boldsymbol{x}_{t-1}\}$. The control signal $\boldsymbol{u}_t$ is based on the difference between the reference input and estimate of the succeeding state vectors $\{\boldsymbol{x}_{t+1}, \cdots, \boldsymbol{x}_{t+q}\}$. Then the state-space equation of CLDS is given by

$$\begin{cases} \boldsymbol{x}_t = \sum_{i=1}^{p} A_i \boldsymbol{x}_{t-i} + A_0 \boldsymbol{u}_t + \boldsymbol{v}_t, \quad \boldsymbol{v}_t \sim \mathcal{N}(0, \Sigma_v) \\ \boldsymbol{u}_t = \sum_{j=1}^{q} D_j \left( \tilde{\boldsymbol{x}}_{\boldsymbol{t+j}} - \sum_{i=1}^{p} A_i \boldsymbol{x}_{t+j-i} \right) \end{cases} \tag{3}$$

After substituting $\boldsymbol{u}_t$, we obtain the following equation:

$$\boldsymbol{x}_t = \sum_{i=-p}^{-1} \Phi_{p+1+i}\boldsymbol{x}_{t+i} + \sum_{j=1}^{q} \Phi_{p+j}\tilde{\boldsymbol{x}}_{t+j} + \boldsymbol{v}_t, \quad \boldsymbol{v}_t \sim \mathcal{N}(0, \Sigma_v) \tag{4}$$

where $\Phi_k$ represents the linear combination of $A_i$ and $D_j$. In the training we replace the reference states $\tilde{\boldsymbol{x}}_{t+j}(j = 1, \ldots, q)$ with $\boldsymbol{x}_{t+j}(j = 1, \ldots, q)$ in Equation (4). Then the state equation (4) becomes a non-causal system, whose parameters can be derived by least squares estimation (LSE) [9].

$$\begin{aligned} [\Phi_1, \Phi_2, \cdots \Phi_{p+q}] &= G_0 \left[ G_{-p}^T, \cdots, G_{-1}^T, G_1^T, \cdots, G_q^T \right]^{-1} \\ \Sigma_v &= \tfrac{1}{N-p-q} \left( R_{0,0} - \sum_{i=-p}^{q} \Phi_{i+p+1} R_{i,0} \right) \end{aligned} \tag{5}$$

where $G_i = [R_{i,-p}, \cdots, R_{i,-1}, R_{i,1}, \cdots, R_{i,q}], -p \leq i \leq q$ and $R_{i,j} = \sum_{t=1-i}^{N-j} \boldsymbol{x}_{t+i}\boldsymbol{x}_{t+j}^T$.

We follow the order determination algorithm described in [16] to choose the appropriate size of the temporal neighborhood $\Omega = p + q$ in Equation (4). In our implementation, the cost function of model fitting is defined as $\sigma_{p+q} = \frac{1}{N-p-q} \sum_{t=p+1}^{N-q} \theta_t^T \theta_t$, where $\theta_t = \boldsymbol{x}_t - \sum_{i=-p}^{-1} \Phi_{p+1+i}\boldsymbol{x}_{t+i} - \sum_{j=1}^{q} \Phi_{p+j}\boldsymbol{x}_{t+j}$. We initialize $p = 1, q = 1$ and then gradually increase $p$ and $q$ respectively until $\frac{\sigma_{p+q+1}}{\sigma_{p+q}} > \mu$.

## 5 Synthesis Using CLDS

It is nontrivial to directly sample a closed-loop system. To sample CLDS, We first sample a reference sequence, and then use the system equation to smooth the discontinuity of the reference sequence. The synthesis algorithm is shown in the Fig. 4.

Using the hidden state vectors $\{\boldsymbol{x}_t\}_{t=1}^N$ generated from the original image sequence $\{I_t\}_{t=1}^N$, we compute the state-to-state similarities and store them in a matrix $\boldsymbol{S} = \langle \boldsymbol{x}_{1:N}, \boldsymbol{x}_{1:N} \rangle = \{s_{i,j}\} (i, j = 1, 2, \ldots, N)$, where $s_{ij} = \frac{\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle}{\sqrt{\langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle \langle \boldsymbol{x}_j, \boldsymbol{x}_j \rangle}}$ and $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ is the inner product of column vectors.

From the similarity matrix $\boldsymbol{S}$, we obtain the state transfer probability matrix $\boldsymbol{P}$ through an exponential function

$$P_{i,j} = P(\boldsymbol{x}_j | \boldsymbol{x}_i) = \begin{cases} \exp\{\gamma s_{i+1,j}\}, & i \neq j \\ 0, & i = j \end{cases}.$$

All the probabilities for any row of $\boldsymbol{P}$ are normalized so that $\sum_j P_{i,j} = 1$. In our implementation, $\gamma$ was set as a number between 1 and 50.

By sampling original state vectors $\{\boldsymbol{x}_i\}_{i=1}^N$, we generate a reference sequence $\{\boldsymbol{x}_{k_1:k_1+\tau-1}, \boldsymbol{x}_{k_2:k_2+\tau-1}, \cdots \boldsymbol{x}_{k_h:k_h+\tau-1}\}$ by concatenating $h$ short state clips

1. To synthesize $M$ frames, set the clip size $\tau$ and select the first clip $\{\boldsymbol{x}_{k_1}, \boldsymbol{x}_{k_1+1}, \cdots, \boldsymbol{x}_{k_1+\tau-1}\}$ randomly from original states $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$
2. Sample the next clip $\{\boldsymbol{x}_{k_2}, \boldsymbol{x}_{k_2+1}, \cdots, \boldsymbol{x}_{k_2+\tau-1}\}$ with $P(\boldsymbol{x}_{k_2}|\boldsymbol{x}_{k_1+\tau-1})$ in $\boldsymbol{P}$
3. Repeat 2, until $h$ state clips $\{\boldsymbol{x}_{k_1:k_1+\tau-1}, \boldsymbol{x}_{k_2:k_2+\tau-1}, \cdots, \boldsymbol{x}_{k_h:k_h+\tau-1}\}$ are sampled. The previous $M(M \leq h \times \tau)$ reference states are used as the initial synthesis states $\{\boldsymbol{x}_1^{(0)}, \boldsymbol{x}_2^{(0)}, \cdots, \boldsymbol{x}_M^{(0)}\}$
4. Iterate $n = 1, 2, \cdots$ until $\left|\delta^{(n)} - \delta^{(n-1)}\right| < \varepsilon$
   a) Sample noise $\boldsymbol{v}_t^{(n)}$ and update $\boldsymbol{x}_t^{(n)}$ by Equation(6) from $t = 1, 2, \cdots, M$
   b) compute iterative error $\delta^{(n)}$
5. Output $\{\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_2^{(n)}, \cdots, \boldsymbol{x}_M^{(n)}\}$

**Fig. 4.** Synthesis algorithm for the closed-loop LDS

such as $\boldsymbol{x}_{k_1:k_1+\tau-1}$ and $\boldsymbol{x}_{k_2:k_2+\tau-1}$. Each clip has $\tau$ frames and two neighboring clips $\boldsymbol{x}_{k_1:k_1+\tau-1}$ and $\boldsymbol{x}_{k_2:k_2+\tau-1}$ are chosen so that the last state $\boldsymbol{x}_{k_1+\tau-1}$ in clip $\boldsymbol{x}_{k_1:k_1+\tau-1}$, and the first state $\boldsymbol{x}_{k_2}$ in clip $\boldsymbol{x}_{k_2:k_2+\tau-1}$, are close according to state-transfer matrix $\boldsymbol{P}$. This is similar, in spirit, to the idea of video texture[11]. However, there exist visible jumps between these short clips in this initial concatenated sequence. We employ an iterative refinement algorithm to successively improve the discontinuity in the output sequence. Specifically, we use the system equation iteratively to smooth out the discontinuity to obtain $\boldsymbol{x}_t$.

$$\boldsymbol{x}_t^{(n)} = \sum_{i=-p}^{-1} \boldsymbol{\Phi}_{p+1+i} \boldsymbol{x}_{t+i}^{(n)} + \sum_{j=1}^{q} \boldsymbol{\Phi}_{p+j} \boldsymbol{x}_{t+j}^{(n-1)} + \boldsymbol{v}_t^{(n)}, \quad \boldsymbol{v}_t^{(n)} \sim \mathcal{N}(0, \Sigma_v) \tag{6}$$

where $\boldsymbol{x}^{(n)}, \boldsymbol{x}^{(n-1)}$ represent the values of $\boldsymbol{x}$ before and after the $n^{th}$ iteration respectively. When the whole state sequence satisfy the statistic process of Equation (4), the sampled values of the whole sequence is unchanged after iteration. Therefore, the $n^{th}$ iterative error can be defined as

$$\delta^{(n)} = \sum_{t=1}^{M} \left\| \boldsymbol{x}_t^{(n)} - \boldsymbol{x}_t^{(n-1)} \right\|_2.$$

The final state sequence is obtained iteratively until the iterative error can hardly drop down. Then we map these states in the low-dimensional space to the observations in the high-dimensional space and obtain the new image sequences.

In Fig. 5, we show the initially concatenated reference video sequence, the intermediate sequence after 5 iterations and the final synthesized sequence. Although jumps are clearly visible between short clips in the initial sequence, CLDS was able to smooth over the whole sequence to produce visually appealing dynamic texture for the SMOKE-NEAR example.

**Fig. 5. Improvement by iteration in synthesis:** (a)-(c) respectively shows 4 frames around a jump before the $1^{st}$ iteration, after 5 iterations and after 30 iterations.

**Table 1.** Comparison of fitting errors of three methods

| Original Sequence | FIRE | SMOKE-FAR | SMOKE-NEAR |
|---|---|---|---|
| Basic LDS method | 55264 | 230.7 | 402.6 |
| Doretto's method | 55421 | 250.0 | 428.2 |
| Our CLDS method | 1170 | 21.4 | 34.4 |

## 6 Experimental Results

We have synthesized many dynamic textures using our proposed CLDS. Our results are compared with those generated using previous nonparametric and parametric methods. All the original sequences are borrowed from the MIT Temporal Textures database[1] and Kwatra et al.'s web site[2]. All the results (videos) can be found in the project's web page[3]. In our experiments, we set the clip size $\tau$ varying from 5 to 20. Through manually choose $\mu$ and $\varepsilon$, the neighborhood size $\Omega$ is between 2 and 4, and iteration number $n$ is between 5 and 50.

As shown in Fig. 7, our approach outperforms not only the noise-driven LDS by Soatto et al[12], but also the improved open-loop LDS by Doretto et al[2] in terms of the visual quality of the synthesis results. For instance, to synthesize dynamic textures for SMOKE-FAR and FIRE sequences, results (Fig. 7(b)(f)) using the basic open-loop LDS algorithm are less likely acceptable because the system is "unstable". Although Doretto et al attempted to solve this problem

[1] ftp://whitechapel.media.mit.edu/pub/szummer/temporal-texture/
[2] http://www.cc.gatech.edu/cpl/projects/graphcuttextures/
[3] http://research.microsoft.com/asia/download/disquisition/dynamictexture(ECCV04_Supp).html

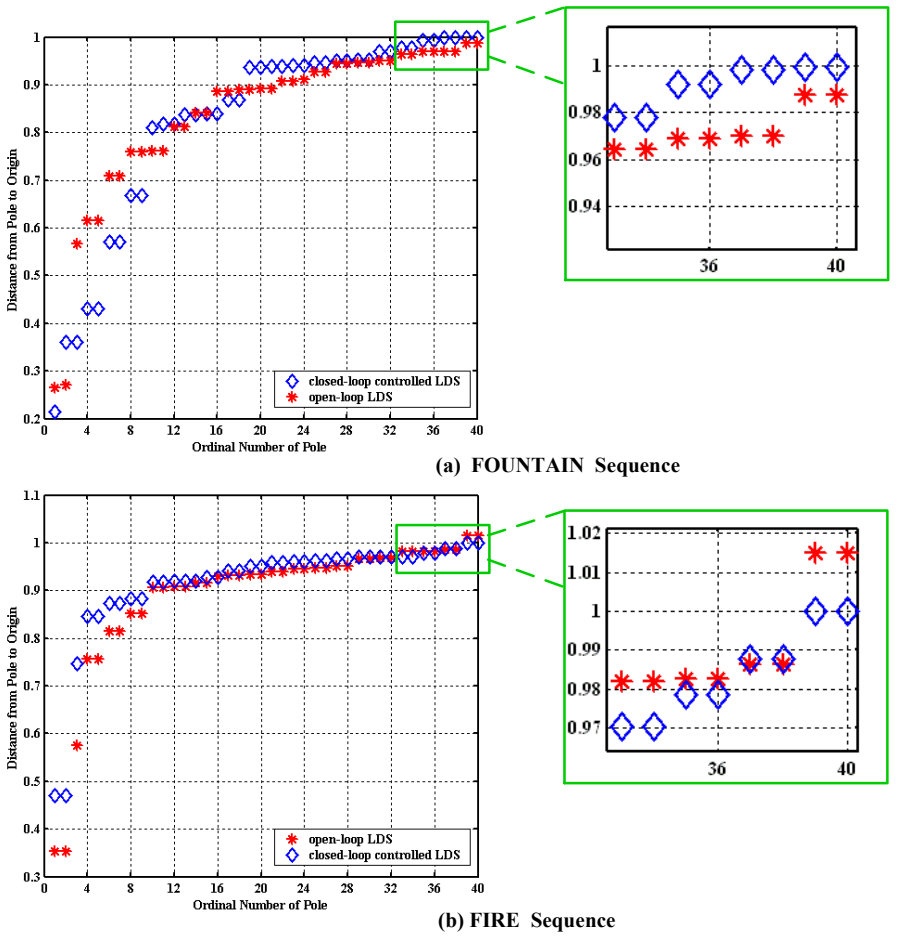



**Fig. 6. Comparison of pole placements between open-loop LDS and closed-loop LDS:** The distance from each pole to the origin is plotted for FOUNTAIN and FIRE sequences. The improvement in pole placement is zoomed in. Some stable poles of FOUNTAIN and all unstable poles of FIRE are relocated onto the unit circle, from which the distance to the origin is 1. With CLDS, poles are moved towards the unit circle, making the system more oscillatory.

by scaling down the poles, the fitting error cannot be ignored and causes that the synthesized video (shown in Fig. 7(c)(g)) still looks different from the original sequence. Our results show that the generated SMOKE-FAR and FIRE sequences closely resemble the original sequences as shown in Fig. 7(d)(h).

Our results also compare favorably with those generated from nonparametric methods. Kwatra[8] has generated perhaps the most impressive dynamic texture synthesis results to date using the graph cut technique. Our results on Bldg9-

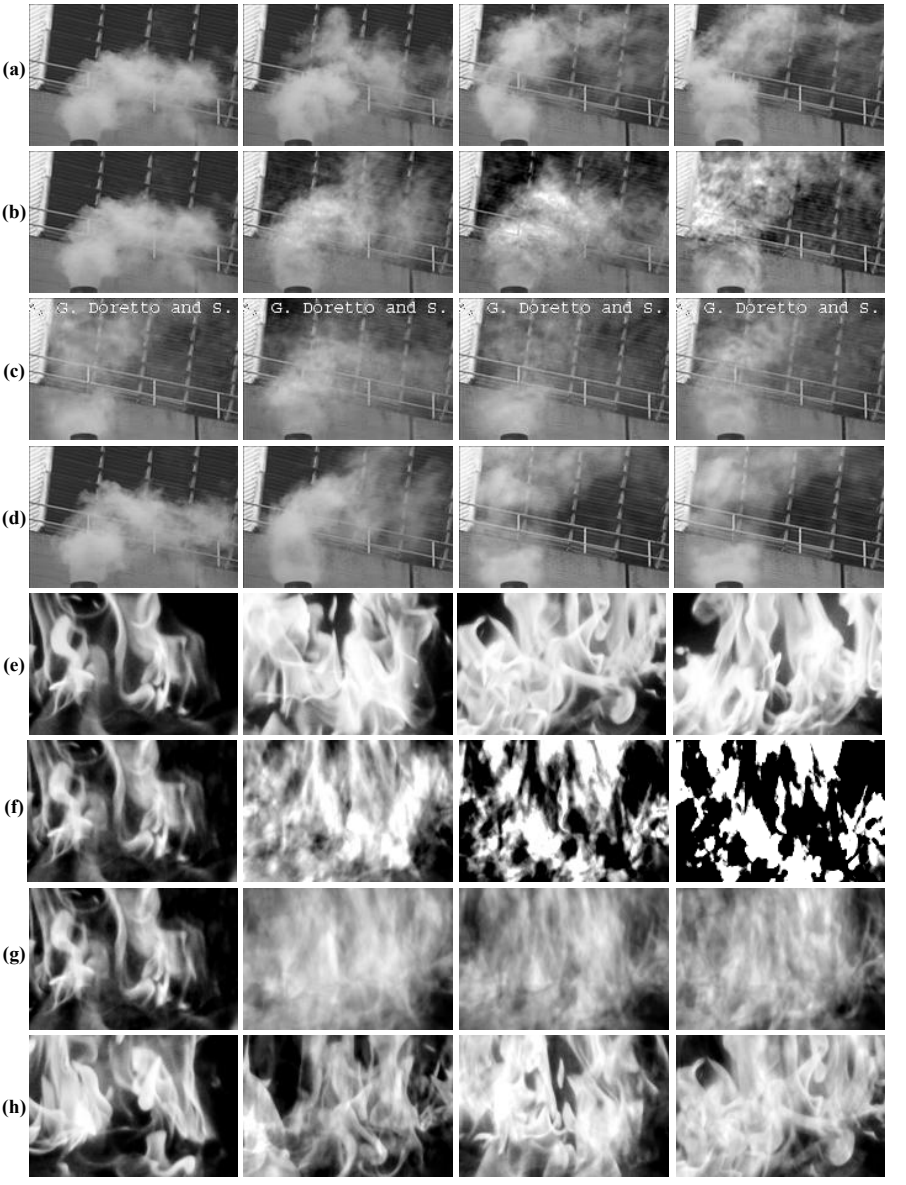



**Fig. 7. Comparison of results between previous LDS methods and ours.** (a) Original SMOKE-FAR sequence (100 frames). (b)-(d) Synthesized SMOKE-FAR sequence (300 frames) respectively by the basic noise-driven LDS, Doretto's method (borrowed from Doretto's web site) and our algorithm. (e) Original FIRE sequence (70 frames). (f)-(h) Synthesized FIRE sequence (300 frames) respectively by the basic noise-driven LDS. , Doretto's method (with our implementation) and our algorithm.

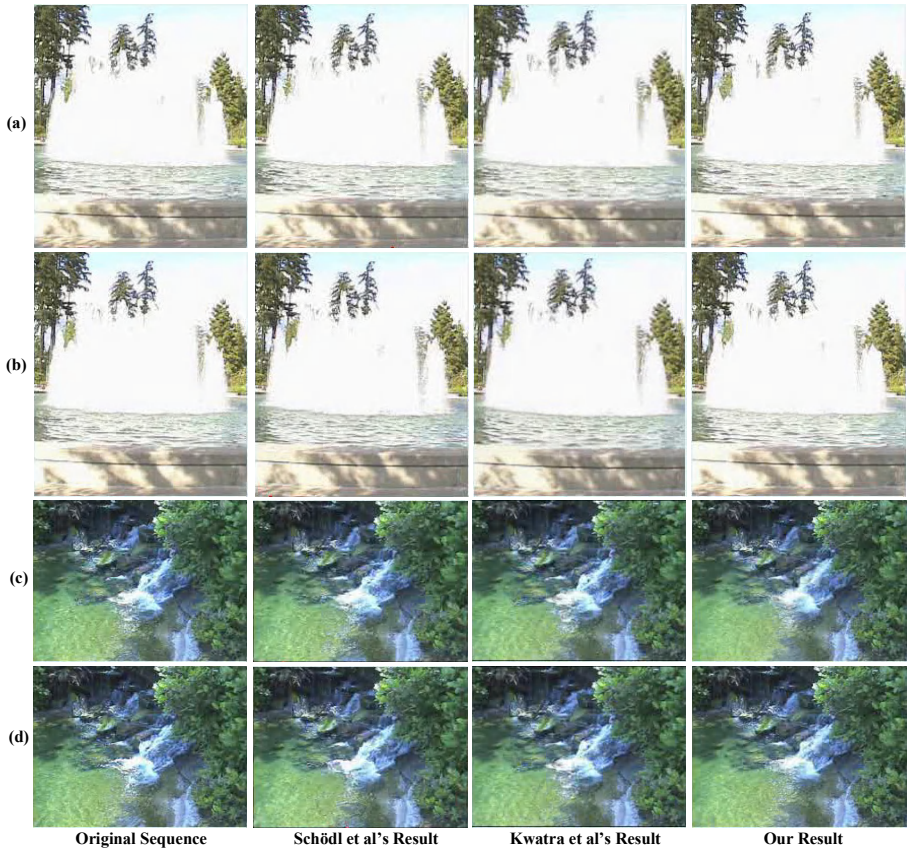



**Fig. 8. Comparison of results between classic nonparametric methods and ours.** From the left column to the right column, it respectively shows the original sequence, the synthesized sequence by Schödl et al's method[11], by Kwatra et al's method[8] and by our method. (a)-(b) are the 93th, 148th frame of Bldg9-FOUNTAIN. (c)-(d) are the 40th, 2085th frame of WATERFALL(the 40th, 87th frame for original video.

FOUNTAIN and WATERFALL shown in Fig. 8 demonstrate that we can achieve similar visual quality using CLDS.

## 7 Discussion

Our experimental results demonstrate that CLDS can produce dynamic texture sequence with good visual quality. There are indeed two problems in the open-loop LDS which are addressed in CLDS, i.e. achieving oscillatory poles and minimizing fitting error.

The effect of CLDS can be observed from how it alters pole placement. Given a synthesized sequence using CLDS, we can compute its effective poles and com-

pare with the poles of the corresponding open-loop LDS. From the FOUNTAIN and FIRE example, we observe significant improvement on pole placement. For the FOUNTAIN sequence shown in Fig. 6(a), most stable poles have been moved towards the unit circle. Two poles are placed exactly on the unit circle, making the whole system oscillatory. For the FIRE sequence shown in Fig. 6(b), two unstable poles from the open-loop LDS have been moved to the unit circle, making it possible to synthesize an infinite sequence of dynamic texture. Note that some stable poles of open-loop LDS close to the origin have been moved closer to the origin with the CLDS. This may cause some blurring in the synthesized results.

Another way to measure the effect of CLDS is to compute the model fitting error. We define the model fitting error as $\delta = \frac{1}{N \times r} \sum_{t=1}^{N} \|\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t\|_2^2$, $(\boldsymbol{x}_t \in \mathbb{R}^r)$, where $\hat{\boldsymbol{x}}_t$ is the estimate of $\boldsymbol{x}_t$. In the basic LDS, $\hat{\boldsymbol{x}}_t = A\boldsymbol{x}_{t-1}$. Doretto[2] relocated unstable poles to the inside of the unit circle. Therefore, the system matrix $A$ is transformed to $\tilde{A}$ and $\hat{\boldsymbol{x}}_t = \tilde{A}\boldsymbol{x}_{t-1}$. For CLDS, instead, $\hat{\boldsymbol{x}}_t = A'\boldsymbol{x}_{t-1} + A'\boldsymbol{u}_t$, where $\boldsymbol{u}_t$ is defined in Equation (2). **Table 1** shows significant improvement in fitting error of CLDS over the basic LDS and Doretto's method. Doretto's pole relocation scheme cannot reduce the model fitting error. Large fitting error implies that the dynamics of the synthesized sequence (Fig. 7(c)(g)) would deviate from the original training set (Fig. 7(a)(e)).

Simple scaling of poles cannot simultaneously satisfy the two goals in generating dynamic texture using the open-loop LDS: making the system oscillatory and minimizing fitting error. On the other hand, CLDS can achieve both goals.

## 8 Summary and Future Work

In this paper, we have analyzed the stability of open-loop LDS used in the dynamic texture model of [12]. We have found that the open-loop LDS must be oscillatory in order to generate an infinite sequence of dynamic texture from a finite sequence of training images. For an LDS that is not oscillatory (stable or unstable), we propose to use feedback control to make it oscillatory. Specifically, we propose a closed-loop LDS (CLDS) to make the system oscillatory and minimize the fitting error. Our experimental results demonstrate our model can be used to synthesize a large variety of dynamic textures with promising visual quality.

In future work, we plan to investigate whether a better controller (e.g. PID or Proportional-Integral-Derivative) would improve the synthesis quality. Another challenging problem is to model and synthesize non-stationary dynamic texture.

## References

1. Z. Bar-Joseph, R. El-Yaniv, D. Lischinski and M. Werman. Texture mixing and texture movie synthesis using statistical learning. *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, pp. 120-135, 2001.

2. G. Doretto, A. Chiuso, Y. N. Wu and S. Soatto. Dynamic Textures. *International Journal of Computer Vision*, vol. 2, pp. 91-109, 2003.
3. G. Doretto, D. Cremers, P. Favaro and S. Soatto. Dynamic Texture Segmentation. *In Proceedings of ICCV'03*, pp. 1236-1242, 2003.
4. G. Doretto and S. Soatto. Editable Dynamic Textures. *In Proceedings of CVPR'03*, pp. 137-142, 2003.
5. A. W. Fitzgibbon. Stochastic rigidity: Image registration for nowhere-static scenes. *In Proceedings of ICCV'01*, vol. 1, pp. 662-670, 2001.
6. G. F. Franklin, J.D. Powell, A. Emami-Naeini. Feedback Control of Dynamic Systems(4th Edition). Prentice Hall, pp. 201-252, 706-797, 2002.
7. B. C. Kuo. Automatic Control Systems(6th Edition). Prentice Hall. pp. 5-15, 1991.
8. V. Kwatra, A. Schödl, I. Essa, G. Turk and A. Bobick. Graphcut Textures: Image and Video Synthesis Using Graph Cuts. *In Proceedings of Siggraph'03*, pp. 277-286, 2003.
9. L. Ljung. System Identification – Theory for the User(2nd Edition). Prentice Hall, 1999.
10. P. Saisan, G. Doretto, Y. N. Wu and S. Soatto. Dynamic Texture Recognition. *Proceedings of CVPR'01*, vol. 2, pp. 58-63, 2001.
11. A. Schödl, R. Szeliski, D. H. Salesin and I. Essa. Video Textures. *In Proceedings of Siggraph'00*, pp. 489-498, 2000.
12. S. Soatto, G. Doretto and Y. N. Wu. Dynamic textures. *In Proceedings of ICCV'01*, vol. 2, pp. 439-446, 2001.
13. M. Szummer and R. W. Picard. Temporal Texture Modeling. *IEEE International Conference on Image Processing*, vol. 3, pp. 823-826, 1996.
14. Y. Z. Wang and S. C. Zhu. A Generative Method for Textured Motion: Analysis and Synthesis. *In Proceedings of ECCV'02*, vol. 1, pp. 583-597, 2002.
15. L. Y. Wei and M. Levoy. Fast Texture Synthesis using Tree-structured Vector Quantization. *In Proceedings of Siggraph'00*, pp. 479-488, 2000.
16. W. W. S. Wei. Time Series Analysis : Univariate and Multivariate Methods. Addison-Wesley, New York. 1990.

# Author Index


Abraham, Isabelle IV-37
Agarwal, Ankur III-54
Agarwal, Sameer II-483
Ahmadyfard, Ali R. IV-342
Ahonen, Timo I-469
Ahuja, Narendra I-508, IV-602
Aloimonos, Yiannis IV-229
Antone, Matthew II-262
Argyros, Antonis A. III-368
Armspach, Jean-Paul III-546
Arnaud, Elise III-302
Arora, Himanshu I-508
Åström, Kalle III-252
Attias, Hagai IV-546
Aubert, Gilles IV-1
Auer, P. II-71
Avidan, Shai IV-428
Avraham, Tamar II-58
Ayache, Nichlas III-79

Bab-Hadiashar, Alireza I-83
Baker, Patrick IV-229
Balch, Tucker IV-279
Bar, Leah II-166
Barnard, Kobus I-350
Bart, Evgeniy II-152
Bartoli, Adrien II-28
Basalamah, Saleh III-417
Basri, Ronen I-574, II-99
Bayerl, Pierre III-158
Bayro-Corrochano, Eduardo I-536
Bebis, George IV-456
Bect, Julien IV-1
Belhumeur, Peter I-146
Belongie, Serge II-483, III-170
Bennamoun, Mohammed II-495
Besserer, Bernard III-264
Bharath, Anil A. I-482, III-417
Bicego, Manuele II-202
Bille, Philip II-313
Bissacco, Alessandro III-456
Blake, Andrew I-428, II-391
Blanc-Féraud, Laure IV-1
Borenstein, Eran III-315
Bouthemy, Patrick III-145
Bowden, Richard I-390
Brady, Michael I-228, I-390
Brand, Matthew II-262
Bretzner, Lars I-322
Bronstein, Alexander M. II-225
Bronstein, Michael M. II-225
Brostow, Gabriel J. III-66
Broszio, Hellward I-523
Brown, M. I-428
Brox, Thomas II-578, IV-25
Bruckstein, Alfred M. III-119
Bruhn, Andrés IV-25, IV-205
Bülow, Thomas III-224
Burger, Martin I-257
Burgeth, Bernhard IV-155
Byvatov, Evgeny II-152

Calway, Andrew II-379
Caputo, Barbara IV-253
Carbonetto, Peter I-350, III-1
Carlsson, Stefan II-518, IV-442
Chai, Jin-xiang IV-573
Chambolle, Antonin IV-1
Charbonnier, Pierre II-341
Charnoz, Arnaud IV-267
Chellappa, Rama I-588
Chen, Chu-Song I-108, II-190
Chen, Jiun-Hung I-108
Chen, Min III-468
Chen, Qian III-521
Chen, Yu-Ting II-190
Cheng, Qiansheng I-121
Chiuso, Alessandro III-456
Christoudias, Chris Mario IV-481
Chua, Chin-Seng III-288
Chung, Albert C.S. II-353
Cipolla, Roberto II-391
Claus, David IV-469
Cohen, Isaac II-126
Cohen, Michael II-238
Comaniciu, Dorin I-336, I-549
Cootes, T.F. IV-316
Coquerelle, Mathieu II-28
Cremers, Daniel IV-74
Cristani, Marco II-202
Cristóbal, Gabriel III-158

Dahmen, Hansjürgen I-614
Dalal, Navneet I-549
Daniilidis, Kostas II-542
Darcourt, Jacques IV-267
Darrell, Trevor IV-481, IV-507
Davis, Larry S. I-175, III-482
Dellaert, Frank III-329, IV-279
Demirci, M. Fatih I-322
Demirdjian, David III-183
Deriche, Rachid II-506, IV-127
Derpanis, Konstantinos G. I-282
Devernay, Frédéric I-495
Dewaele, Guillaume I-495
Dickinson, Sven I-322
Doretto, Gianfranco II-591
Dovgard, Roman II-99
Drew, Mark S. III-582
Drummond, Tom II-566
Duan, Ye III-238
Duin, Robert P.W. I-562
Dunagan, B. IV-507
Duraiswami, Ramani III-482

Ebner, Marc III-276
Eklundh, Jan-Olof IV-253, IV-366
Engbers, Erik A. III-392
Eong, Kah-Guan Au II-139
Eriksson, Martin IV-442
Essa, Irfan III-66

Fagerström, Daniel IV-494
Faugeras, Olivier II-506, IV-127, IV-141
Favaro, Paolo I-257
Feddern, Christian IV-155
Fei, Huang III-497
Fergus, Robert I-242
Fermüller, Cornelia III-405
Ferrari, Vittorio I-40
Finlayson, Graham D. III-582
Fischer, Sylvain III-158
Fitzgibbon, Andrew W. IV-469
Freitas, Nando de I-28, I-350, III-1
Freixenet, Jordi II-250
Fritz, Mario IV-253
Frolova, Darya I-574
Frome, Andrea III-224
Fua, Pascal II-405, II-566, III-92
Fuh, Chiou-Shann I-402
Furukawa, Yasutaka II-287
Fussenegger, M. II-71

Gavrila, Darin M. IV-241
Gheissari, Niloofar I-83
Ghodsi, Ali IV-519
Giblin, Peter II-313, II-530
Giebel, Jan IV-241
Ginneken, Bram van I-562
Goldlücke, Bastian II-366
Gool, Luc Van I-40
Grossauer, Harald II-214
Gumerov, Nail III-482
Gupta, Rakesh I-215
Guskov, Igor I-133
Gyaourova, Aglika IV-456

Hadid, Abdenour I-469
Haider, Christoph IV-560
Hanbury, Allan IV-560
Hancock, Edwin R. III-13, IV-114
Hartley, Richard I. I-363
Hayman, Eric IV-253
Heinrich, Christian III-546
Heitz, Fabrice III-546
Herda, Lorna II-405
Hershey, John IV-546
Hertzmann, Aaron II-299, II-457
Hidović, Džena IV-414
Ho, Jeffrey I-456
Ho, Purdy III-430
Hoey, Jesse III-26
Hofer, Michael I-297, IV-560
Hong, Byung-Woo IV-87
Hong, Wei III-533
Horaud, Radu I-495
Hsu, Wynne II-139
Hu, Yuxiao I-121
Hu, Zhanyi I-190, I-442
Huang, Fay II-190
Huang, Jiayuan IV-519
Huber, Daniel III-224

Ieng, Sio-Song II-341
Ikeda, Sei II-326
Irani, Michal II-434, IV-328

Jacobs, David W. I-588, IV-217
Jawahar, C.V. IV-168
Je, Changsoo I-95
Ji, Hui III-405
Jia, Jiaya III-342
Jin, Hailin II-114

Jin, Jesse S. I-270
Johansen, P. IV-180
Jones, Eagle II-591
Joshi, Shantanu III-570

Kadir, Timor I-228, I-390
Kaess, Michael III-329
Kanade, Takeo III-558, IV-573
Kanatani, Kenichi I-310
Kang, Sing Bing II-274
Kasturi, Rangachar IV-390
Kervrann, Charles III-132
Keselman, Yakov I-322
Khan, Zia IV-279
Kimia, Benjamin II-530
Kimmel, Ron II-225
Kiryati, Nahum II-166, IV-50
Kittler, Josef IV-342
Kohlberger, Timo IV-205
Kokkinos, Iasonas II-506
Kolluri, Ravi III-224
Koulibaly, Pierre Malick IV-267
Koudelka, Melissa I-146
Kriegman, David I-456, II-287, II-483
Krishnan, Arun I-549
Krishnan, Sriram I-336
Kristjansson, Trausti IV-546
Kück, Hendrik III-1
Kuijper, Arjan II-313
Kumar, Pankaj I-376
Kumar, R. III-442
Kuthirummal, Sujit IV-168
Kwatra, Vivek III-66
Kwolek, Bogdan IV-192

Lagrange, Jean Michel IV-37
Lee, Kuang-chih I-456
Lee, Mong Li II-139
Lee, Mun Wai II-126
Lee, Sang Wook I-95
Lenglet, Christophe IV-127
Leung, Thomas I-203
Levin, Anat I-602, IV-377
Lhuillier, Maxime I-163
Lim, Jongwoo I-456, II-470
Lim, Joo-Hwee I-270
Lin, Stephen II-274
Lin, Yen-Yu I-402
Lindenbaum, Michael II-58, III-392, IV-217
Lingrand, Diane IV-267
Little, James J. I-28, III-26
Liu, Ce II-603
Liu, Tyng-Luh I-402
Liu, Xiuwen III-570, IV-62
Lladó, Xavier II-250
Loog, Marco I-562, IV-14
López-Franco, Carlos I-536
Lourakis, Manolis I.A. III-368
Lowe, David G. I-28
Loy, Gareth IV-442
Lu, Cheng III-582

Ma, Yi I-1, III-533
Magnor, Marcus II-366
Maire, Michael I-55
Malik, Jitendra III-224
Mallick, Satya P. II-483
Mallot, Hanspeter A. I-614
Manay, Siddharth IV-87
Manduchi, Roberto IV-402
Maragos, Petros II-506
Marsland, S. IV-316
Martí, Joan II-250
Matei, B. III-442
Matsushita, Yasuyuki II-274
Maurer, Jr., Calvin R. III-596
McKenna, Stephen J. IV-291
McMillan, Leonard II-14
McRobbie, Donald III-417
Medioni, Gérard IV-588
Meltzer, Jason I-215
Mémin, Etienne III-302
Mendonça, Paulo R.S. II-554
Mian, Ajmal S. II-495
Mikolajczyk, Krystian I-69
Miller, James II-554
Mio, Washington III-570, IV-62
Mittal, Anurag I-175
Montagnat, Johan IV-267
Mordohai, Philippos IV-588
Moreels, Pierre I-55
Morency, Louis-Philippe IV-481
Moreno, Pedro III-430
Moses, Yael IV-428
Moses, Yoram IV-428
Muñoz, Xavier II-250
Murino, Vittorio II-202

Narayanan, P.J. IV-168
Nechyba, Michael C. II-178

Neumann, Heiko III-158
Ng, Jeffrey I-482
Nguyen, Hieu T. II-446
Nicolau, Stéphane III-79
Nielsen, Mads II-313, IV-180
Nillius, Peter IV-366
Nir, Tal III-119
Nistér, David II-41
Noblet, Vincent III-546

Odehnal, Boris I-297
Okada, Kazunori I-549
Okuma, Kenji I-28
Oliensis, John IV-531
Olsen, Ole Fogh II-313
Opelt, A. II-71
Osadchy, Margarita IV-217
Owens, Robyn II-495

Padfield, Dirk II-554
Pallawala, P.M.D.S. II-139
Papenberg, Nils IV-25
Paris, Sylvain I-163
Park, JinHyeong IV-390
Park, Rae-Hong I-95
Pavlidis, Ioannis IV-456
Peleg, Shmuel IV-377
Pelillo, Marcello IV-414
Pennec, Xavier III-79
Perez, Patrick I-428
Perona, Pietro I-55, I-242, III-468
Petrović, Vladimir III-380
Pietikäinen, Matti I-469
Pinz, A. II-71
Piriou, Gwenaëlle III-145
Pollefeys, Marc III-509
Pollitt, Anthony II-530
Ponce, Jean II-287
Pottmann, Helmut I-297, IV-560
Prados, Emmanuel IV-141

Qin, Hong III-238
Qiu, Huaijun IV-114
Quan, Long I-163

Rahimi, A. IV-507
Ramalingam, Srikumar II-1
Ramamoorthi, Ravi I-146
Ranganath, Surendra I-376
Redondo, Rafael III-158
Reid, Ian III-497
Ricketts, Ian W. IV-291
Riklin-Raviv, Tammy IV-50
Roberts, Timothy J. IV-291
Rohlfing, Torsten III-596
Rosenhahn, Bodo I-414
Ross, David II-470
Rother, Carsten I-428
Russakoff, Daniel B. III-596

Saisan, Payam III-456
Samaras, Dimitris III-238
Sarel, Bernard IV-328
Sato, Tomokazu II-326
Satoh, Shin'ichi III-210
Savarese, Silvio III-468
Sawhney, H.S. III-442
Schaffalitzky, Frederik I-363, II-41, II-85
Schmid, Cordelia I-69
Schnörr, Christoph IV-74, IV-205, IV-241
Schuurmans, Dale IV-519
Seitz, Steven M. II-457
Sengupta, Kuntal I-376
Sethi, Amit II-287
Shahrokni, Ali II-566
Shan, Y. III-442
Shashua, Amnon III-39
Shokoufandeh, Ali I-322
Shum, Heung-Yeung II-274, II-603, III-342
Simakov, Denis I-574
Singh, Maneesh I-508
Sinha, Sudipta III-509
Sivic, Josef II-85
Smeulders, Arnold W.M. II-446, III-392
Smith, K. IV-316
Soatto, Stefano I-215, I-257, II-114, II-591, III-456, IV-87
Sochen, Nir II-166, IV-50, IV-74
Soler, Luc III-79
Sommer, Gerald I-414
Sorgi, Lorenzo II-542
Spacek, Libor IV-354
Spira, Alon II-225
Srivastava, Anuj III-570, IV-62
Steedly, Drew III-66
Steiner, Tibor IV-560
Stewenius, Henrik III-252
Stürzl, Wolfgang I-614
Sturm, Peter II-1, II-28

Sugaya, Yasuyuki I-310
Sullivan, Josephine IV-442
Sun, Jian III-342
Suter, David III-107
Szepesvári, Csaba I-16

Taleghani, Ali I-28
Tang, Chi-Keung II-419, III-342
Tarel, Jean-Philippe II-341
Taylor, C.J. IV-316
Teller, Seth II-262
Thiesson, Bo II-238
Thiré, Cedric III-264
Thormählen, Thorsten I-523
Thureson, Johan II-518
Todorovic, Sinisa II-178
Tomasi, Carlo III-596
Torma, Péter I-16
Torr, Philip I-428
Torresani, Lorenzo II-299
Torsello, Andrea III-13, IV-414
Treuille, Adrien II-457
Triggs, Bill III-54, IV-100
Tsin, Yanghai III-558
Tsotsos, John K. I-282
Tu, Zhuowen III-195
Turek, Matt II-554
Tuytelaars, Tinne I-40
Twining, C.J. IV-316

Ullman, Shimon II-152, III-315
Urtasun, Raquel II-405, III-92

Vasconcelos, Nuno III-430
Vemuri, Baba C. IV-304
Vidal, René I-1

Wada, Toshikazu III-521
Wallner, Johannes I-297
Wang, Hanzi III-107
Wang, Jue II-238
Wang, Zhizhou IV-304
Weber, Martin II-391
Weickert, Joachim II-578, IV-25, IV-155, IV-205
Weimin, Huang I-376
Weiss, Yair I-602, IV-377
Weissenfeld, Axel I-523
Welk, Martin IV-155
Wen, Fang II-603
Wildes, Richard P. I-282
Wills, Josh III-170
Windridge, David I-390
Wolf, Lior III-39
Wong, Wilbur C.K. II-353
Wu, Fuchao I-190
Wu, Haiyuan III-521
Wu, Tai-Pang II-419
Wu, Yihong I-190

Xiao, Jing IV-573
Xu, Ning IV-602
Xu, Yingqing II-238
Xydeas, Costas III-380

Yan, Shuicheng I-121
Yang, Liu III-238
Yang, Ming-Hsuan I-215, I-456, II-470
Yao, Annie II-379
Yao, Jian-Feng III-145
Yezzi, Anthony J. II-114, IV-87
Ying, Xianghua I-442
Yokoya, Naokazu II-326
Yu, Hongchuan III-288
Yu, Jingyi II-14
Yu, Simon C.H. II-353
Yu, Tianli IV-602
Yu, Yizhou III-533
Yuan, Lu II-603
Yuille, Alan L. III-195

Zandifar, Ali III-482
Zboinski, Rafal III-329
Zelnik-Manor, Lihi II-434
Zeng, Gang I-163
Zha, Hongyuan IV-390
Zhang, Benyu I-121
Zhang, Hongjiang I-121
Zhang, Ruofei III-355
Zhang, Zhongfei (Mark) III-355
Zhou, S. Kevin I-588
Zhou, Xiang Sean I-336
Zhu, Haijiang I-190
Zisserman, Andrew I-69, I-228, I-242, I-390, II-85
Zomet, Assaf IV-377